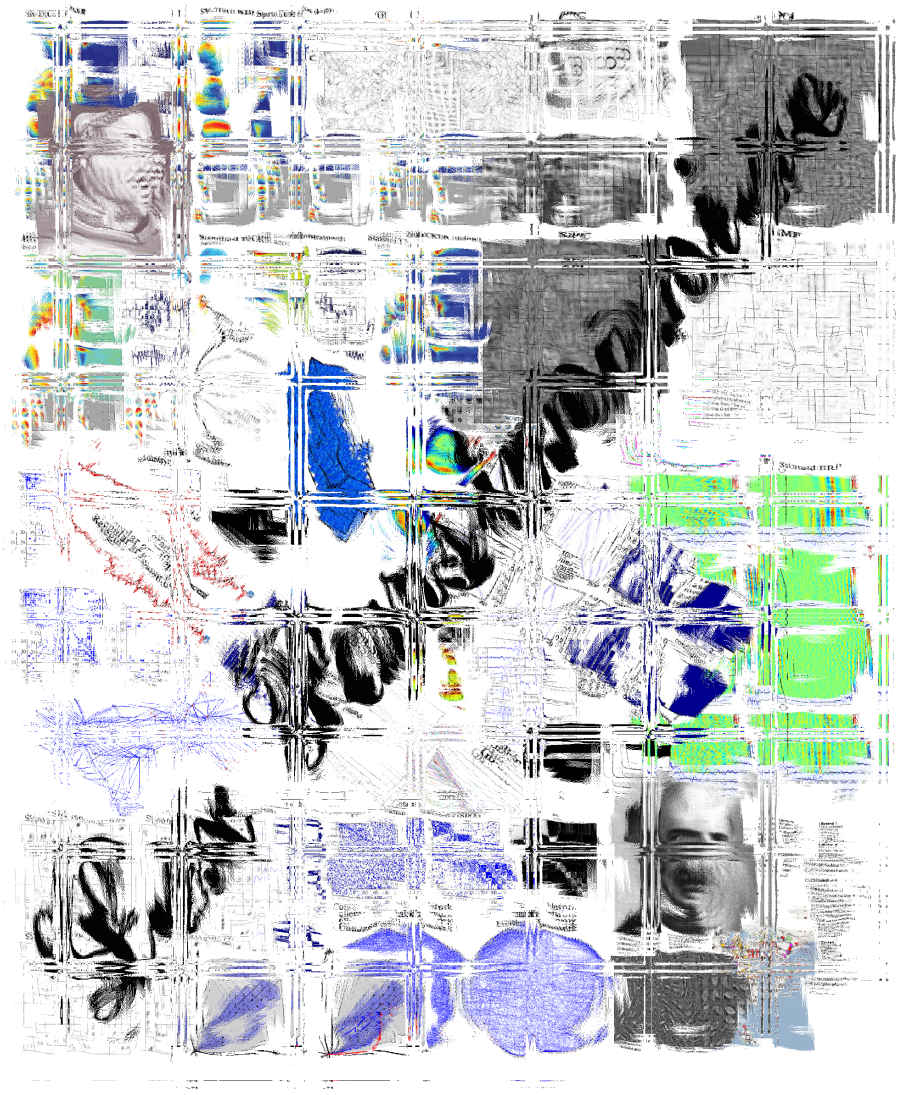
Decomposition methods for unsupervised learning

Mørup, Morten

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

Decomposition Methods for Unsupervised Learning



Morten Mørup

Kongens Lyngby 2008
IMM-PHD-2008-194

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Preface

This thesis was prepared at the Department of Informatics and Mathematical Modelling at the Technical University of Denmark (DTU), in partial fulfillment of the requirements for acquiring a Ph.D. degree in mathematical modelling.

The thesis deals with the application and development of decomposition methods for Unsupervised Learning. The thesis consists of a review of various decomposition methods used in Unsupervised Learning namely matrix and tensor decomposition methods, decomposition under invariance constraints such as shift, reverberation and rotation and clustering methods based on decomposition.

Part of the research was conducted at the Department of Computer Science at Stanford University as well as at the Mathematics Department at Berkeley University, USA.

The project was supervised by Professor Lars Kai Hansen (DTU Informatics) and partly supervised by Ph.D. Dr. Med. Sidse Marie Arnfred as well as Associate Professor Ole Winther. Funding was provided by the Technical University of Denmark.

The thesis is printed by DTU Informatics and available as softcopy from http://www.imm.dtu.dk.

Kgs. Lyngby, April 2008

Morten Mørup

Acknowledgement

The motto of the search engine Google Scholar is "Stand on the shoulders of giants". To write this PhD I have myself stood on the shoulders of many whom I wish to thank.

My deepest gratitude is extended towards my principal mentors over the course of this project. These include my supervisor Lars Kai Hansen (IMM) who opened my eyes to the exciting world of Machine Learning and made sure my research always was going in a fruitful direction. I thank you for your great support and I am grateful to have been able to benefit from your deep understanding yet great overview of Machine Learning methods. Sidse Marie Arnfred who introduced me to the interesting field of Electroencephalography (EEG) research - I am thankful for your mentorship as well as your friendship. Ole Winther and Jan Larsen who inspired me through their teaching and always kept an open door for discussing problems.

A turning point for my PhD was an early trip to the Workshop on Tensor Decomposition and Application in Luminy, France. There I among many great people meet Brett Bader, Gene H. Golub, Richard Harshman, Tammy G. Kolda and Lek-Heng Lim all of whom have inspired and encouraged me and my research. Research can sometimes be a lonesome walk but the openness and enthusiasm from these people made the work enjoyable. I wish in particular to thank Gene H. Golub who gave me the opportunity to visit Stanford for half a year the Summer and Fall 2006. I wish to thank Richard Harshman for taking me under his wing and inviting me to the TRICAP meeting in Crete. I also want to thank Lek-Heng Lim for taking good care of me both during my visit at Stanford as well as my visit at Berkeley's department of Mathematics in the Fall 2007. I

further wish to thank Tammy and Brett for great discussions and friendship and for giving me the opportunity to present my work at the ICIAM conference in Zurich the Summer of 2007.

From Copenhagen University life sciences I wish to thank Giorgio Tomasi for helping to explain important aspects of tensor computation and Rasmus Bro for his enthusiasm and openness to discuss problems.

I further want to thank all my colleagues at IMM. In particular, I wish to thank Kristoffer Hougaard Madsen for great collaboration and help as well as Mikkel Nørgaard Schmidt and Line Harder Clemmensen. I have truly enjoyed working together with you and benefitted greatly from our many discussions. I also want to thank the department secretary Ulla Nørhave for all her help on practical details.

Unfortunately, both Gene Golub and Richard Harshman passed away during the final months of my PhD. I was saddened by the loss of such great friends and mentors but very grateful I got the chance to get them both to know.

(Gene H. Golub, 1932-2007)

I wish to give a special thank to Gene Golub who gave me the opportunity to visit Stanford for half a year. Gene, you made Stanford a second home for both Erika and I. You greatly inspired me both through your deep insights but also through your worldliness and enthusiasm.

(Richard A. Harshman, 1943-2008)

A special thank I wish to give Richard Harshman. His many ideas has served as a great inspiration for me and I am honored he took me under his wings and introduced me to the tensor community.

Finally, I wish to thank my wife Erika for her great support, for choosing to live with me in Denmark and for making the lonely PhD work so easy to cope with.

Abstract

This thesis presents the application and development of decomposition methods for Unsupervised Learning. It covers topics from classical factor analysis based decomposition and its variants such as Independent Component Analysis, Non-negative Matrix Factorization and Sparse Coding to their generalizations to multi-way array, i.e. tensor decomposition, through models such as the CanDecomp/PARAFAC and the Tucker model. Extensions for these types of decomposition models to incorporate shift, reverberation and general transformations are also described. Finally, a connection between decomposition methods and clustering problems is derived both in terms of classical point clustering but also in terms of community detection in complex networks. A guiding principle throughout this thesis is the principle of parsimony. Hence, the goal of Unsupervised Learning is here posed as striving for simplicity in the decompositions. Thus, it is demonstrated how a wide range of decomposition methods explicitly or implicitly strive to attain this goal. Applications of the derived decompositions are given ranging from multi-media analysis of image and sound data, analysis of biomedical data such as electroencephalography to the analysis of social network data.

Resumé

Denne afhandling præsenterer anvendelser og udvikling af dekompositionsmetoder for Unsupervised Learning (indlæring uden supervision). Fra klassisk faktor-analyse baserede dekompositioner med relaterede modeller såsom Independent Component Analysis, Non-negative Matrix Factorization og Sparse Coding til generaliseringer af disse modeller til multi-vejs data, dvs. tensordekompositioner, ved modeller såsom CandeComp/PARAFAC og Tucker modellen. Udvidelser af disse typer af dekompositioner, der kan håndtere tidsforskydninger, ekko-effekter og mere generelle transformationer omhandles også. Endelig udledes en forbindelse mellem dekompositionsmetoder og gruppering af data (clustering) både med hensyn til klassisk geometrisk gruppering men også med hensyn til detektering af grupper i komplekse netværk. Et vigtigt underliggende princip for Unsupervised Learning og et gennemgående tema for denne afhandling er at opnå simple repræsentationer. Således demonstreres det, hvorledes en lang række dekompositionsmetoder eksplicit eller implicit søger mod dette mål. Anvendelser af de udledte dekompositionsmetoder gives både indenfor multi-medie analyse af billede og lyd, analyse af biomedicinske signaler såsom elektroencefalografi samt analyse af sociale netværk.

List of Published Papers during the PhD

Journal Papers

- S.M. Arnfred, L.K. Hansen, J. Parnas, M. Mørup, Regularity Increases Middle-latency Evoked and Late Induced Beta Brain Response Following Proprioceptive Stimulation, accepted for publication, Brain Research, 2008
- M. Mørup, L. K. Hansen, S. M. Arnfred, Algorithms for Sparse Nonnegative Tucker, accepted for publication, Neural Computation, 2008
- I. Griskova, M. Mørup, J. Parnas, O. Ruksenas, S. Arnfred, The amplitude and phase precision of 40 Hz auditory steady-state response depend on the level of arousal, Experimental Brain Research, vol. 183(1), pp. 133-138, 2007
- M. Mørup, L.K. Hansen, S. M. Arnfred, ERPWAVELAB A toolbox for multi-channel analysis of time-frequency transformed event related potentials, Journal of Neuroscience Methods, vol. 161, pp. 361-368, 2007
- S.M. Arnfred, L.K. Hansen, J. Parnas, M. Mørup, Proprioceptive Evoked Gamma Oscillations, Brain Research, vol. 1147, pp. 167-174, 2007
- M. Mørup, L. K. Hansen, C. S. Hermann, J. Parnas, S. M. Arnfred, Parallel Factor Analysis as an exploratory tool for wavelet transformed event-related EEG, NeuroImage, vol. 29(3), pp. 938-947, 2006

Conference Papers

- M. Mørup, K.H. Madsen, L.K. Hansen, Approximate L0 constrained Non-negative Matrix and Tensor Factorization, accepted ISCAS 2008 special session on Non-negative Matrix and Tensor Factorization and Related Problems
- P. M. Rasmussen, M. Mørup, L. K. Hansen, S. M. Arnfred, Model Order Estimation for Independent Component Analysis of Epoched EEG Signals, Biosignals 2008 International Conference on Bio-inspired Systems and Signal Processing., 2008
- M. Mørup, L. H. Clemmensen, Multiplicative updates for the LASSO, Machine Learning for Signal Processing, 2007 IEEE Workshop on, pages 33-38, 2007
- M. Mørup, K. H. Madsen, L. K. Hansen, Shifted Non-negative Matrix Factorization, Machine Learning for Signal Processing, 2007 IEEE Workshop on, pages 139-144, 2007
- M. Mørup, K. H. Madsen, L. K. Hansen, Shifted Independent Component Analysis, Independent Component Analysis and Blind Signal Separation, pages 89-96, 2007
- M. N. Schmidt, M. Mørup, Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation, Independent Component Analysis and Blind Signal Separation, pages 700-707, 2006

Submitted Papers

- M. Mørup, M. N. Schmidt, Efficient Rotation and Shift Invariant Sparse Coding, Submitted, 2008
- M. Mørup, L.K. Hansen, L-H. Lim, Simplicial Clustering, Submitted, 2008
- M. Mørup, L.K. Hansen, S.M. Arnfred, L.-K. Lim, K.M. Madsen , Shift Invariant Multilinear Decomposition of Neuroimaging Data, submitted NeuroImage 2008
- M. Mørup, M. N. Schmidt, L. K. Hansen, Shift Invariant Sparse Coding of Image and Music Data, submitted, Journal of Machine Learning Research (JMLR), 2007
- K.H. Madsen, L.K. Hansen, M. Mørup, Combining Time and Frequency Domain Optimization: Shifts, Convolution and Smoothness in Factor Analysis Type Decompositions, submitted, Journal of Signal Processing Systems special issue on MLSP2007

Toolbox

- www.erpwavelab.org - a software for the analysis and visualization of wavelet transformed event related EEG Data.

Technical Reports

- M. N. Schmidt, M. Mørup, Sparse Non-negative Matrix Factor 2-D Deconvolution for Automatic Transcription of Polyphonic Music, 2006
- M. Mørup, L. K. Hansen, J. Parnas, S. M. Arnfred, Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization, 2006
- M. Mørup, M. N. Schmidt, Sparse Non-negative Tensor 2D Deconvolution (SNTF2D) for multi channel time-frequency analysis, 2006

Contents

Notation

$\mathcal{X}$	Tensor or multi-way array of more than 2 modalities
$\mathcal{X}^{I\times J\times K}$	Tensor or multi-way array of size $I \times J \times K$
$\boldsymbol{X}$	A matrix or 2-way array
$\boldsymbol{X}^{I\times J}$	A matrix or 2-way array of size $I \times J$
$\boldsymbol{x}_j$	The j^{th} column of $\boldsymbol{X}$
$\boldsymbol{x}_{i,:}$	The i^{th} row of $\boldsymbol{X}$
$\boldsymbol{X}_{i,j}$	The element of the i^{th} row and j^{th} column of $\boldsymbol{X}$
$\boldsymbol{X} \in \mathbb{R}_+^{I\times J}$	the matrix $\boldsymbol{X}^{I\times J}$ is real valued and non-negavite
$\mathcal{I}$	Diagonal tensor
$\boldsymbol{I}$	Diagonal matrix
$\boldsymbol{A}^{(n)}$	A matrix indexed by n
$\boldsymbol{X}_{(n)}$	The n-mode matricizing: $\mathcal{X}^{I_1\times I_2\times\ldots I_N} \rightarrow \boldsymbol{X}^{I_n\times I_1 I_2\cdots I_{n-1}I_{n+1}\cdots I_N}$
$\boldsymbol{A}\otimes\boldsymbol{B}$	The Kronecker product, i.e. $\boldsymbol{A}\otimes\boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & \cdots & a_{1,J}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{I,1}\boldsymbol{B} & \cdots & a_{I,J}\boldsymbol{B} \end{bmatrix}$
$\boldsymbol{A}\odot\boldsymbol{B}$	The Khatri-Rao product $\boldsymbol{A}\odot\boldsymbol{B} = [\boldsymbol{a}_1\otimes\boldsymbol{b}_1\ \boldsymbol{a}_2\otimes\boldsymbol{b}_2\ \ldots\ \boldsymbol{a}_J\otimes\boldsymbol{b}_J]$
$\boldsymbol{A}\bullet\boldsymbol{B}$	Elementwise multiplication, i.e. $(\boldsymbol{A}\bullet\boldsymbol{B})_{i,j} = \boldsymbol{A}_{i,j}\cdot\boldsymbol{B}_{i,j}$
$\frac{\boldsymbol{A}}{\boldsymbol{B}}$	Elementwise division, i.e. $(\frac{\boldsymbol{A}}{\boldsymbol{B}})_{i,j} = \boldsymbol{A}_{i,j}/\boldsymbol{B}_{i,j}$
$e^{\boldsymbol{A}}$	Elementwise exponential, i.e. $(e^{\boldsymbol{A}})_{i,j} = e^{\boldsymbol{A}_{i,j}}$
$\boldsymbol{A}^{\cdot\alpha}$	Elementwise exponentiation, i.e. $(\boldsymbol{A}^{\cdot\alpha})_{i,j} = \boldsymbol{A}_{i,j}^{\alpha}$
$\boldsymbol{A}\geq 0$	$\boldsymbol{A}_{i,j}\geq$ for all i,j similarly for $\leq, <, >, =$
$\nabla_{\boldsymbol{A}}g(u)$	Elementwise differentiation, i.e. $(\nabla_{\boldsymbol{A}}g(u))_{i,j} = \frac{\partial g(u)}{\partial \boldsymbol{A}_{i,j}}$
$\times_n$	$(\mathcal{Q}\times_n\mathbf{P})_{i_1,i_2,\ldots,j_n,\ldots i_N} = \sum_{i_n}\mathcal{Q}_{i_1,i_2,\ldots,i_n,\ldots i_N}\mathbf{P}_{j_n,i_n}$
$\|\boldsymbol{X}\|_p$	the l_p norm, $\|\boldsymbol{X}\|_p := \begin{cases} p>0\ (\sum_{i,j}\lvert\boldsymbol{X}_{i,j}\rvert^p)^{1/p} \\ p=0\ \#\{i,j \mid \boldsymbol{X}_{i,j}\neq 0\} \\ p\rightarrow\infty\ \max_{i,j}\lvert\boldsymbol{X}_{i,j}\rvert \end{cases}$
	($\|\boldsymbol{X}\|_p$ is only a true norm for $p\geq 1$)
$\|\boldsymbol{X}\|_F$	The Frobenius norm,i.e. $\|\boldsymbol{X}\|_F = \sqrt{\sum_{i,j}\boldsymbol{X}_{i,j}^2}$
$\mathcal{F}(\mathbf{u})$	The fourier transform of $\mathbf{u}$, i.e. $\widetilde{u}_f = \sum_{j=0}^{J-1}u_j e^{-i2\pi\frac{f}{J}j}$
$\mathcal{F}^{-1}(\widetilde{\mathbf{u}})$	The inverse fourier transform of $\widetilde{\mathbf{u}}$, i.e. $u_j = \frac{1}{J}\sum_{f=0}^{J-1}\widetilde{u}_f e^{i2\pi\frac{f}{J}j}$
$\widetilde{\mathbf{u}}$	$\mathcal{F}(\mathbf{u})$
$\widetilde{\boldsymbol{U}}$	$\mathcal{F}(\boldsymbol{U})$
$\widetilde{\mathcal{U}}$	$\mathcal{F}(\mathcal{U})$
$\widetilde{\boldsymbol{U}}^H$	Conjugate transpose of $\widetilde{\boldsymbol{U}}$
$\widetilde{\mathbf{U}}^{(f)}$	Complex matrix given by $\boldsymbol{U}\bullet e^{-i2\pi\frac{f}{J}\boldsymbol{\tau}}$
$\boldsymbol{S}^{\dagger}$	The pseudo-inverse of $\boldsymbol{S}$ given by $\boldsymbol{S}^{\top}(\boldsymbol{S}\boldsymbol{S}^{\top})^{-1}$
$[\boldsymbol{A}]^+$	$\boldsymbol{A}$ with any negative elements set to zero
$[\boldsymbol{A}]^-$	$[\boldsymbol{A}]^- = [-\boldsymbol{A}]^+$
$E[X]$	The expected value of X

Abbreviations

AI	Artificial Intelligence
AIC	Akaike's Information Criterion
BIC	Bayesian Information Criterion
BPD	Basis Pursuit Denoising
BSS	Blind Source Separation
CandeComp	Canonical Decomposition
CC	Cross-Correlation procedure
cCP	Convolutive CP
cICA	Convolutive ICA
cNMF	Convolutive NMF
CP	CandeComp/PARAFAC
cSC	Convolutive Sparse Coding
DEDICOM	Decomposition into Directional Components
DFT	Discrete Fourier Tranform
EEG	Electroencephalography
fMRI	Functional Magnetic Resonance Imaging
GB-SC	Gradient Based Sparse Coding
HOSVD	Higher Order Singular Value Decomposition
ICA	Independent Component Analysis
IDFT	The Inverse Discrete Fourier Tranform
i.i.d.	independent and identically distributed
ITPC	Inter Trial Phase Coherence
LAD	Least Absolute Deviation
LARS	Least Angle Regression and Selection
LASSO	Least Absolute Shrinkage and Selection Operator
MU	Multiplicative Updates
NLARS	Non-negative LARS
NMF	Non-negative Matrix Factorization
NR	Newton-Raphson procedure
NTF	Non-negatived Tensor Factorization
ONMF	Orthogonal NMF
PARAFAC	Parallel Factor Analysis
PCA	Principal Component Analysis
PMF	Positive Matrix Factorization
PTF	Positive Tensor Factorization
RSISC	Rotation and Shift Invariant Sparse Coding
SC	Sparse Coding
SGC	Simplicial Graph Clustering
ShiftNMF	Shifted Non-negative Matrix Factorization
ShiftSC	Shifted Sparse Coding
SICA	Shifted Independent Component Analysis
SISA	Shift Invariant Subspace Analysis
SISC	Shift Invariant Sparse Coding
SNMF	Sparse Non-negative Matrix Factorization

SN-TUCKER	Sparse Non-negative Tucker
SPC	Simplicial Point Clustering
SPCA	Sparse Principal Component Analysis
SVD	Singular Value Decomposition
TFGM	Time-Frequency Gradient Method
Tucker	The Tucker model
Tucker2	The Tucker model where loadings of one mode set to identity

Chapter 1

Introduction

Machine Learning is a rapidly growing field devoted to the formal study of learning systems. The field is highly interdisciplinary and as such borrows and builds upon ideas from statistics, computer science, engineering, neuroscience, optimization, biology, psychology and physics. Machine Learning is related to the field of Artificial Intelligence (AI), however, while AI is motivated by human beings and attempts to simulate them with the help of computers - Machine Learning is mainly motivated by the capabilities of modern computers. Hence the goal in Machine Learning is to explore all possible routs to create automated learning as well as their applications. Machine Learning can roughly be separated into the following three types of learning problems

- **Supervised Learning:** A technique for learning a function from a set of training data. The goal is to make the learned function generalize from the presented data to unseen situations in a "reasonable" way. The training data consist of pairs of input as well as desired outputs.

- **Reinforcement Learning:** A technique for learning how an agent ought to take actions in an environment so as to maximize some notion of long-term reward. Reinforcement Learning differs from the Supervised Learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected only rewards given for the actions the agent takes.

- **Unsupervised Learning:** Unsupervised Learning is distinguished from supervised learning by the fact that there is no a priori output. In Unsupervised Learning, a data set only of input objects is gathered. Unsupervised learning then typically attempts to account for this data set by some more simplistic representation.

This thesis will focus on Unsupervised Learning, namely, decomposition methods for these types of learning problem.

Much of our intelligence, in particular our perception, is acquired without a teacher, hence, Unsupervised Learning is an important learning strategy in the development of our brains. Just through mere exposure humans and animals learn how to analyze their environments and recognize relevant objects and events. Thus, Unsupervised Learning is not only an important type of learning problem in Machine Learning but also an important aspect of Artificial Intelligence.

For our brain to learn unsupervised there must be some guiding principle or goal. However, what is this goal? What is it we want a system to learn without giving external instruction? As Wang [2001] points out there is no simple answer to this crucial question. In their tutorial on Unsupervised Learning Ghahramani and Roweis [1999] define the purpose of Unsupervised Learning to be:

- Perform dimensionality reduction
- Build topographic maps
- Find the hidden causes or sources of the data
- Model the data density
- Cluster data

Beneath these objectives is the fundamental task of representation - Unsupervised Learning attempts to derive hidden structure from the raw data. This is a meaningful endeavor as pointed out by Wang [2001] because input data are far from random but are produced by physical processes. As such, in their book on Unsupervised Learning Hinton and Sejnowski [1999] define the purpose of Unsupervised Learning as *to extract an efficient internal representation of the statistical structure implicit in the inputs.*

The goal of finding an efficient internal representation of data has become a highly relevant task due to the massive amount of data that has arisen due to our ever increasing ability to store as well as analyze huge quantities of information.

Figure 1.1: Examples of massive modern data sets. From the left a micro-array recording of DNA sequences, a 3 tesla Siemens MRI scanner, term-document occurrence matrix and a visualization of the link structure of the Internet.

I.e., the storage capacity and computational power has followed Moore's law and doubled approximately every second years since the 1970's. These massive amount of data arise in multiple of areas of research ranging from

- Bioinformatics (for instance DNA or protein microarray data and protein interaction network data)
- Neuroinformatics (for instance data obtained from functional Magnetic Resonance Imaging (fMRI), Electroencephalography (EEG) and Positron Emission Tomography (PET))
- Web data mining (for instance term-document occurrence data, graphs based on links between web pages, social networks as well as recorded user behavior)

to mention but a few. For all these data the ability to extract an efficient internal representation is crucial in finding the hidden causes as well as underlying structure in the data. As such Unsupervised Learning has become an important and fascinating field of study that facilitates the analysis of these large modern data sets. Not only do the methods used in Unsupervised Learning help find the hidden causes and structure of the data, but the way in which information is extracted also open doorways to the understanding of how we human process large quantities of information.

This thesis is devoted to decomposition methods for Unsupervised Learning. Decomposition, as referred to in this thesis, is the process of finding hidden internal representation of the data, i.e., to decompose the data into internal representations. The guiding principle for an efficient internal representation will be simplicity of the representation. As is demonstrated in the next chapter this aim has many names. Among many are perhaps the most well-known terms parsimony, redundancy reduction and sparsity. Despite Wang [2001] pointing out that there is no simple answer to the goal of Unsupervised Learning, this

thesis will be completely biased towards a formulation of Unsupervised Learning based on the goal of simplicity of the extracted internal representation.

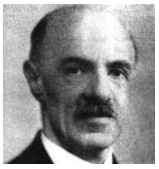
The thesis is structure as follows

- Chapter 2 describes factor analysis based decomposition which are examples of some of the most fundamental and classic decomposition models for Unsupervised Learning.
- Chapter 3 describes the extension of factor analysis based decomposition to multi-way arrays, i.e. tensors. In particular to the CandeComp/PARAFAC and Tucker model.
- Chapter 4 describes extensions of the aforementioned decompositions to incorporate invariance. The main emphasis of the chapter will be invariance to time delays and general shifts in the data.
- Chapter 5 will emphasize the relevance of decomposition methods for clustering problems. Both in terms of classical geometric clustering problems but also for community detection in graphs.

Chapter 2

Factor Analysis based Decomposition

"It is the faith of all science that an unlimited number of phenomena can be comprehended in terms of a limited number of concepts or ideal constructs. Without this faith no science could ever have any motivation." -Thurstone



Charles Spearman
(1863-1945)

Charles Spearman the father of factor analysis started his psychology career relatively late in life, beginning his Ph.D. studies at age 34 and accepting his first university position at the age of 42. He was the first to offer a tenable psychometric definition of intelligence, and is therefore considered to be the father of classical test theory [Jensen, 1994]. In his famous article, *"General Intelligence Objectively Determined and Measured"* [Spearman, 1904], Spearman proposed the idea that intelligent behavior is generated by a single, unitary quality within the human mind or brain. Spearman derived this theoretical entity, called the general factor, or simply g, through a new statistical technique called factor analysis (`http://www.indiana.edu/~intell/spearman.shtml`). Although proponents of multiple intelligence theory reject his proposed g quantity, factor analysis remains one of the most important tools not only in 21st century intelligence research but also for science in general.

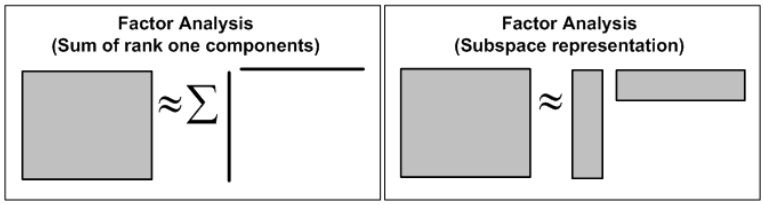


Figure 2.1: Two common graphical representations of the factor analysis model. To the left, the model is given as a sum of rank one components. To the right, the data $\boldsymbol{X}$ as well as the complete matrices $\boldsymbol{A}$ and $\boldsymbol{S}$ are given. In general $D < \min\{I, J\}$ such that the factor analysis model forms a subspace representation of the data.

The factor analysis model Spearman derived can be written as

$$\boldsymbol{X}^{I\times J} \approx \boldsymbol{A}^{I\times D}\boldsymbol{S}^{D\times J}. \tag{2.1}$$

The data matrix, $\boldsymbol{X}$, Spearman considered was given by the scores obtained by different subjects from various tests. Thus, $\boldsymbol{X}_{i,j}$ denoted test score of subject j in intelligence test i. As such, the internal representation formed by the model states that the subjects intelligences can be described by D underlying factors such that test i is reflected in the d^{th} factor to degree $\boldsymbol{A}_{i,d}$. While the d^{th} factor pertain to subject j with strength $\boldsymbol{S}_{d,j}$. In general $D < \min\{I, J\}$

Despite the above simple interpretation of the representation the model has one major flaw. The underlying factors are not uniquely defined. Thus the alternative model given by the invertible matrix $\boldsymbol{Q}^{D\times D}$

$$\widehat{\boldsymbol{A}} = (\boldsymbol{A}\boldsymbol{Q}^{-1}), \quad \widehat{\boldsymbol{S}} = (\boldsymbol{Q}\boldsymbol{S}), \tag{2.2}$$

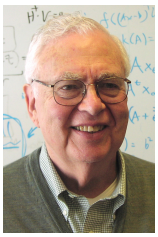
will give an equally good representation, i.e. $\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{S} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{S}}$. To resolve this issue, additional constraints have to be imposed. Deriving constraints that well suits the data $\boldsymbol{X}$ such that $\boldsymbol{A}$ and $\boldsymbol{S}$ are unique has been the focus of many researchers ever since Spearman proposed his factor analysis model.

2.1 Singular Value Decomposition and Principal Component Analysis

The Singular Value Decomposition (SVD) of a matrix $\boldsymbol{X}$ is given by the decomposition that solves the problem

$$\begin{aligned} \text{minimize}_{\boldsymbol{U},\boldsymbol{\Sigma},\boldsymbol{V}} \quad & \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\|_F^2 \\ \text{subjected to} \quad & \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I} \\ & \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I} \\ & \boldsymbol{\Sigma} \text{ is a diagonal matrix such that } \Sigma_{1,1} \geq \Sigma_{2,2} \ldots \geq \Sigma_{D,D} \end{aligned}$$

The solution can be obtained by solving a sequence of eigenvalue problems of $\boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{X}^\top\boldsymbol{X}$, [Golub and Van Loan, 1996]. The diagonal of $\boldsymbol{\Sigma}$ contains the singular values ordered such that the largest attainable value comes first. The decomposition is unambiguous if the eigenvalues are unique. This is achieved by constraining the solutions in such a way that each consecutive basis vector of the column and row space, i.e. $\mathbf{u}_d$ and $\mathbf{v}_d$ account for as much of the remaining variation when subtracting previous found vectors. The SVD is sometimes also referred to as Principal Component Analysis (PCA).


Gene H. Golub (1932-2007)

Gene H. Golub was a very influential person in the numerical analysis community. He and Kahan presented an efficient algorithm for calculating the SVD [Golub and Kahan, 1965] and with Charles Van Loan he authored the famous book Matrix Computation [Golub and Van Loan, 1996]. Gene dedicated most of his life to the analysis of matrices and as such his license plate could be no other than "PROF SVD".

Although the SVD is generally unique (i.e., as long as each eigenvalue is distinct), the decomposition most often does not yield the desired internal representation of the data. However, the decomposition given by the SVD forms a good starting point for resolving the generally ambiguous representation in the factor analysis model as the method finds the subspace that accounts for as much of the variation in the data possible. Thus, $\boldsymbol{A} = \boldsymbol{U}^{I\times D}\boldsymbol{\Sigma}^{D\times D}$ and $\boldsymbol{S} = \boldsymbol{V}^{\top D\times J}$ can be used to solve for the general ambiguity $\boldsymbol{Q}$.

2.2 Quartimax, Varimax and Orthomax Rotation

Louis L. Thurstone (1887-1955)

In 1947 Thurstone wrote [Thurstone, 1947]

> *"In a factor problem one is concerned about how to account for the observed correlations among all the variables in terms of the smallest number of factors and with the smallest possible residual error."*

Inspired by this, a variety of criteria for rotating the factor solutions were proposed. In 1953 Carrol [Carroll, 1953] proposed the Quartimax criterion, i.e., maximizing the following quantity

$$C_{Carrol} = -\sum_{d>d'}\sum_{j} \boldsymbol{S}_{d,j}^2 \boldsymbol{S}_{d',j}^2 \tag{2.3}$$

Thus, the rotation matrix $\boldsymbol{Q}$ is to be estimated such that the off diagonal elements in the covariance matrix of the squared elements $\boldsymbol{S}^{\cdot 2}\boldsymbol{S}^{\cdot 2^\top}$ is minimized. In particular, this criterion requires that a large loading of one factor be opposite small loadings of the remaining factors. Shortly after Carrol proposed his Quartimax criterion the following similar criterions were suggested by Neuhaus and Wrigley [1954]; Saunders [1953]; Ferguson [1954] respectively

$$\begin{aligned}
C_{Neuhaus} &= \frac{1}{JD}\sum_j\sum_d \boldsymbol{S}_{d,j}^4 - \frac{1}{J^2D^2}\left(\sum_j\sum_d \boldsymbol{S}_{d,j}^2\right)^2 \\
C_{Saunders} &= JD\frac{\sum_j\sum_d \boldsymbol{S}_{d,j}^4}{\left(\sum_j\sum_d \boldsymbol{S}_{d,j}^2\right)^2} \\
C_{Ferguson} &= \sum_j\sum_d \boldsymbol{S}_{d,j}^4
\end{aligned}$$

While Neuhaus criterion maximizes the total variance of the squared matrix $\boldsymbol{S}^{\cdot 2}$, Saunders approach requires the fourth moment over second moment squared, i.e. the kurtosis, to be maximal. Fergusons criterion simply seek to maximize the fourth moment irrespective of the second moment structure. The Varimax criterion proposed by Kaiser [Kaiser, 1958] is given by

$$C_{Kaiser} = \sum_d(\sum_j \boldsymbol{S}_{d,j}^4 - \frac{1}{J}(\sum_j \boldsymbol{S}_{d,j}^2)^2),$$

i.e., maximizing the variance of the squared loadings component-wise rather than by the full matrix. Essentially, the aim of all the above criterions for factor

rotation is simplicity or parsimony. That is to attain loadings with small or large scores such that the factors pertain to specific aspects of the original data. A simple extension of the above criteria form the Orthomax rotation criterion [Crawford and Ferguson, 1970]

$$C_{Orthomax} = \sum_d (\sum_j \boldsymbol{S}_{d,j}^4 - \frac{\gamma}{J}(\sum_j \boldsymbol{S}_{d,j}^2)^2)$$

where γ interpolates between Quartimax (as defined by Ferguson) $\gamma = 0$ and Varimax $\gamma = 1$. Rotation of factors have been a very active field of research within Psychometry and a whole range of related criteria to the above such as the Equimax and Parsimax [Crawford and Ferguson, 1970] have been proposed. However, the goal of all the criteria are the same – to account for the data in a way that is the most parsimonious.

2.3 Independent Component Analysis

The goal of Independent Component Analysis (ICA) is to minimize the statistical dependence of the components of the representation. Since most of the research on ICA has the application of source separation in mind, many authors treating the ICA problem do not use the term ICA, but speak simply of blind source separation (BSS). Just as for the Varimax and Quartimax method in the previous section, the estimation of the data model of Independent Component Analysis is usually performed by maximizing a socalled contrast function, i.e., an objective function measuring independence. The literature on ICA is extensive, however, excellent reviews on the subject can be found in [Hyvarinen et al., 2001; Amari and A., 1998; Cardoso, 1998; Hyvärinen, 1999; Lee et al., 2000].

A common approach to ICA is the maximum likelihood (ML) method [Pham and Garrat, 1997; Hyvarinen et al., 2001] which corresponds to the approach of maximizing information proposed in [Bell and Sejnowski, 1995]. In the framework of ML a non-gaussian source distribution is assumed such that ambiguity can be resolved up to the trivial ambiguities of scale and permutation by maximizing the likelihood of the sources given this source distribution. Stated in terms of learning in a neural network [Bell and Sejnowski, 1995] the basic problem to be tackled is how to maximize the mutual information, I, that the output $\boldsymbol{Y}$ of a neural network processor contains about its input $\boldsymbol{S}$. This is defined in terms of entropy, $H(x) = -E[\log x]$,

$$I(\boldsymbol{Y}, \boldsymbol{S}) = H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{S})$$

where $H(\boldsymbol{Y}|\boldsymbol{S})$ is whatever entropy the output has that does not depend on the input. This term can be neglected since the relation under consideration between input and output is deterministic. Thus, the mutual information can be

maximized by maximizing the entropy of the output alone [Bell and Sejnowski, 1995]. Let $\boldsymbol{U} = \boldsymbol{QS}$. Consider the non-linear function $\boldsymbol{Y}_{i,j} = g(\boldsymbol{U}_{i,j})$. Assuming $\boldsymbol{S}$ is i.i.d. the multivariate probability density function is then given by

$$\prod_j f_{\mathbf{y}_j}(\mathbf{y}_j) = \prod_j \frac{f_{\mathbf{s}_j}(\mathbf{s}_j)}{|\boldsymbol{J}|}$$

where $\boldsymbol{J}$ is the jacobian of the derivative of $\mathbf{y}_j$ with respect to $\mathbf{s}_j$. We now have

$$H(\mathbf{y}_j) = -E\left[ln f_{\mathbf{y}_j}(\mathbf{y}_j)\right]) = E\left[ln|\boldsymbol{J}|\right] - E\left[ln f_{\mathbf{s}_j}(\mathbf{s}_j)\right].$$

Where the second term is independent of $\boldsymbol{Q}$. Maximizing the entropy (i.e. minimizing the neg-entropy) is equivalent to maximizing the likelihood of the sources [Cardoso, 1997] when considering the non-linearity $g(\mathbf{u}_j)$ a cumulative distribution functions of the source distribution $f(\mathbf{u_j})$. By applying the change of variable principle the log-likelihood of the observed ambiguous variables $\mathbf{s}_j$ is given by

$$\log L = \sum_j \log f(\boldsymbol{Q}\mathbf{s}_j) + J \log|\det(\boldsymbol{Q})|. \tag{2.4}$$

where $\boldsymbol{S}$ is regarded a "training set" where each observation is equally likely. Notice how the two terms of equation (2.4) penalizes deviation from source distribution and non-invertibility of $\boldsymbol{Q}$ respectively. (The latter stems from the fact that $\det(\boldsymbol{Q})$ measures the volume of $\boldsymbol{Q}$ and becomes zero if $\boldsymbol{Q}$ is rank deficient). This non-gaussian source distribution could for instance be given by the Laplace distribution $f(\mathbf{u}_j) \propto e^{-\|\mathbf{u}_j\|_1}$ corresponding to a rotation of the sources such that the l_1-norm, i.e. $\|\boldsymbol{U}\|_1$, is as small as possible.

Another common approach to ICA is to jointly diagonalize some higher order moments, cumulants [Comon, 1994] or autocorrelation functions of $\boldsymbol{S}$ [Molgedey and Schuster, 1994]. Finally, some ICA approaches measure deviation from normality for instance through the kurtosis [Comon, 1994] given by

$$kurt(\boldsymbol{S}) = E[\boldsymbol{S}^{\cdot 4}] - 3E[\boldsymbol{S}^{\cdot 2}]^2$$

For Gaussian random variables the kurtosis is zero. It is typically positive for distributions with heavy tails and a peak at zero, and negative for densities with lighter tails, see also figure 2.4. Distributions of positive (resp. negative) kurtosis are thus called super-Gaussian (resp. sub-Gaussian) [Hyvärinen, 1999]. Thus, maximizing this objective will yield sparse representations such that the elements of $\boldsymbol{S}$ is either small or large. Maximizing the kurtosis above corresponds to maximizing $C_{Neuhaus}$. Furthermore, dividing the kurtosis by the square variance of $\boldsymbol{S}$ we get,

$$\frac{kurt(\boldsymbol{S})}{\sigma_{\boldsymbol{S}}^4} = \frac{E[\boldsymbol{S}.^4]}{\sigma_{\boldsymbol{S}}^4} - 3$$

this is also referred to as the excess kurtosis. Maximizing this quantity is equivalent to maximizing $C_{Saunders}$. Thus, this approach to ICA is equivalent to the Quartimax criterions of the previous section. As such, Ferguson motivated his Quartimax criterion already in 1954 [Ferguson, 1954] by writing

> *"It may appear farfetched that the concept of entropy in statistical mechanics or the concept of information employed by Shannon in communication theory should have direct relevance to the rotational problem. An attempt will be made to demonstrate that concepts of this type can be used in reformulating the rotational problem in such a way as to admit the possibility of an acceptable objective solution."*

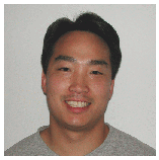
while Comon [1994] entitled his paper on ICA *"Independent component analysis, a new concept?"*.

2.4 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is given by the decomposition

$$\mathbf{X} \approx \mathbf{AS},$$

where $\mathbf{X} \in \mathbb{R}_+^{I \times J}$, $\mathbf{A} \in \mathbb{R}_+^{I \times D}$ and $\mathbf{S} \in \mathbb{R}_+^{D \times J}$, i.e. such that the variables $\mathbf{X}$, $\mathbf{A}$ and $\mathbf{S}$ are non-negative. The decomposition is useful as it results in easy interpretable part based representations [Lee and Seung, 1999]. As Lee and Seung writes in the abstract to their Nature paper on NMF



Daniel D. Lee

Sebastian Seung

"Is perception of the whole based on perception of its parts? There is psychological and physiological evidence for parts-based representations in the brain, and certain computational theories of object recognition rely on such representations. But little is known about how brains or computers might learn the parts of objects. Here we demonstrate an algorithm for non-negative matrix factorization that is able to learn parts of faces and semantic features of text. This is in contrast to other methods, such as principal components analysis and vector quantization, that learn holistic, not parts-based, representations. Non-negative matrix factorization is distinguished from the other methods by its use of non-negativity constraints. These constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. When non-negative matrix factorization is implemented as a neural network, parts-based representations emerge by virtue of two properties: the firing rates of neurons are never negative and synaptic strengths do not change sign."

Non-negative Matrix Factorization is also named Positive Matrix Factorization (PMF) [Paatero and Tapper, 1994] and dates at least back to the work on positive rank decomposition (the p-rank) [Berman and Plemmons, 1979; Gregory and Pullman, 1983; Cohen and Rothblum, 1993]. The decomposition was popularized by Lee and Seung due to a simple algorithmic procedure based on multiplicative updates (MU) [Lee and Seung, 2000], see Box 2.4. The decomposition has proven useful for a wide range of data where non-negativity is a natural constraint. These encompass data for text-mining based on counts, image data, biomedical data [Lee et al., 2001] and spectral data [Gobinet et al., 2004; Pauca et al., 2006]. The decomposition can also be useful even when the data in itself is negative by considering the amplitude of a spectral representation [Smaragdis and Brown, 2003; Schmidt and Mørup, 2006; Mørup et al., 2007a]. Despite constraining the data and decomposition to the positive orthant the decomposition is not in general unique [Donoho and Stodden, 2004; Laurberg et al., 2007]. The model simply states that the data is spanned by the convex hull formed by $[\mathbf{0}\ \boldsymbol{A}]$ (by appropriate normalization). Thus, if the data $\boldsymbol{X}$ resides well inside the positive orthant another representations $[\mathbf{0}\ \widehat{\boldsymbol{A}}]$ will also adequately form a convex hull for the data, see figure 2.2. To alleviate this ambiguity sparseness is often imposed [Hoyer, 2002, 2004; Eggert and Körner, 2004; Mørup et al., 2008d]

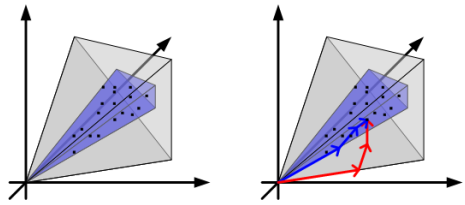


Figure 2.2: The NMF decomposition is not in general unique. For instance, if the data resides well inside the positive orthant the data can be spanned arbitrarily by any set of vectors that form a convex hull for the complete set of data. To resolve ambiguity sparseness is often imposed on one of the modes [Hoyer, 2002, 2004; Eggert and Körner, 2004]. By imposing sparseness, by the l_1-norm the sum of the coefficients of the basis vectors is minimized. Hence, the span given by the tighter convex hull in blue has a lower cost than the span of the larger convex hull given in red.

Box 2.4: Multiplicative Updates (MU)

Let $C(\boldsymbol{\theta})$ be a given objective function of the non-negative variable $\boldsymbol{\theta}$. Let further $\nabla_{\boldsymbol{\theta}} C = [\nabla_{\boldsymbol{\theta}} C]^+ - [\nabla_{\boldsymbol{\theta}} C]^-$ where $[\nabla_{\boldsymbol{\theta}} C]^+$ and $[\nabla_{\boldsymbol{\theta}} C]^-$ denotes positive and negative parts of the gradient respectively. Then the multiplicative update is defined by :

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \bullet \left(\frac{[\nabla_{\boldsymbol{\theta}} C]^-}{[\nabla_{\boldsymbol{\theta}} C]^+} \right)$$

Since all quantities are non-negative, $\boldsymbol{\theta}$ will remain in the positive orthant. A small constant $\varepsilon = 10^{-9}$ can be added to the denominator to avoid potential division by zero. By also adding the constant to the numerator the corresponding gradient is unaltered. When the gradient is zero $[\frac{\partial C(\theta)}{\partial \theta_{i,j}}]^+ = [\frac{\partial C(\theta)}{\partial \theta_{i,j}}]^-$ such that θ is left unchanged. If the gradient is positive $[\frac{\partial C(\theta)}{\partial \theta_{i,j}}] > [\frac{\partial C(\theta)}{\partial \theta_{i,j}}]^-$ hence $\theta_{i,j}$ will decrease and vice versa if the gradient is negative. Thus, there is a one-to-one relation between fixed points of the multiplicative update rule and stationary points under gradient descend.

Motivation of MU: In [Højen-Sørensen et al., 2002] the MU was motivated considering the lagrange multiplier $\boldsymbol{L}$ enforcing non-negativity. The modified cost function including this Lagrange multiplier is $C^{(mod)} = C - \mathrm{tr}(\boldsymbol{L}^\top \boldsymbol{\theta})$. Solving for the Lagrange multiplier we get $\boldsymbol{L} = \nabla_{\boldsymbol{\theta}} C$ The Kuhn-Tucker condition requires $L_{i,j}\theta_{i,j} = (\nabla_{\boldsymbol{\theta}} C)_{i,j}\theta_{i,j} = 0$, hence, solving this equation the multiplicative updates can be derived.

Proof of convergence: The convergence of multiplicative updates have been proven by the use of auxiliary functions [Lee and Seung, 2000; Ding et al., 2006a]: An auxiliary function G to the function C is defined by: $G(\boldsymbol{\theta}, \boldsymbol{\theta}^t) \geq C(\boldsymbol{\theta})$ and $G(\boldsymbol{\theta}, \boldsymbol{\theta}) = C(\boldsymbol{\theta})$. If G is an auxiliary function then C is non-increasing under the update $\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} G(\boldsymbol{\theta}, \boldsymbol{\theta}^t)$. Constructing appropriate auxiliary functions is in general difficult.

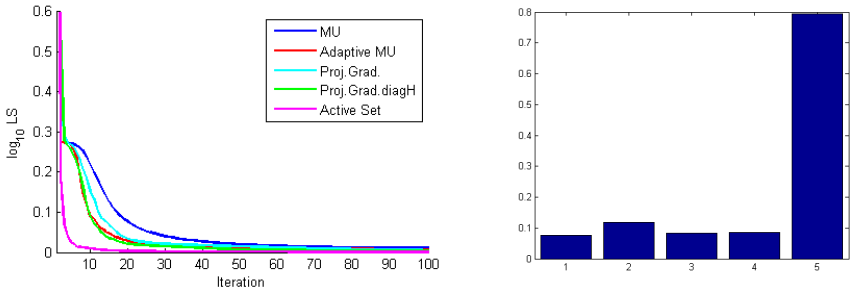


Figure 2.3: **Left panel:** Comparison of 5 different NMF algorithm for least squares minimization analyzing the USPS handwritten digits dataset of size 7291×256 using 9 components. MU is the algorithm formed by multiplicative updates for NMF as proposed in [Lee and Seung, 2000]. Adaptive MU is the multiplicative updates with overrelaxed bound optimization proposed in [Salakhutdinov et al., 2003]. Proj.Grad is the projected gradient method proposed in [Lin, 2007]. Proj.Grad.diagH is the projected gradient method including the diagonal Hessian. Active Set is the active set algorithm described in [Lawson and Hanson, 1974; Bro and de Jong, 1997] and given in algorithm 2.4. Clearly, the MU suffer from slow convergence, however, the two projected gradient method as well as the Adaptive MU converges somewhat better. Despite these improvements, the Active Set by far converges faster than the other algorithms as it finds the optimal $\boldsymbol{A}$ for fixed $\boldsymbol{S}$ and vice versa in each iteration. **Right panel:** Average time in seconds to perform one iteration for each of the five algorithms. Clearly, the MU, Adaptive MU, Proj.Grad and Proj.Grad.diagH takes about the same amount of time whereas the Active Set algorithm is an order of 10 times as expensive as the other algorithms.

In particular, NMF has become a popular type of decomposition since the representation generally results in easily interpreted representations. In the model, the whole becomes the sum of its parts and as such no cancellation between components can hamper the interpretability. Despite that multiplicative updates have become by far the most popular approach to solving for NMF the updates are known to suffer from slow convergence, see figure 2.3. The benefit is that the updates form a simple algorithmic framework that can be used for a large variety of objective functions [Dhillon and Sra, 2006; Cichocki et al., 2006, 2007] while auxiliary functions have been derived that guarantee monotonic decrease using the multiplicative updates [Lee and Seung, 2000; Ding et al., 2006a; Mørup and Clemmensen, 2007]. As such MU was proven in [Lee and Seung, 2000] to monotonically decrease the least squares error.

To alleviate the slow convergence of the multiplicative updates 'over relaxed bound optimization' was proposed in [Salakhutdinov et al., 2003]. Here, a step

size parameter α is tuned forming the overrelaxed update

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} \bullet \left(\frac{[\nabla_{\boldsymbol{\theta}} C]^-}{[\nabla_{\boldsymbol{\theta}} C]^+} \right)^{.\alpha} . \tag{2.5}$$

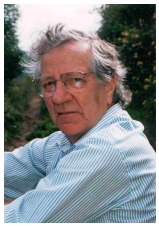
This update can also ensure convergence since when $\alpha \to 0$ only infinitesimal steps in the negative gradient direction are taken. In [Lin, 2007] a projective gradient method was derived based on a gradient based search where negative elements were projected to the positive orthant. Despite these efforts to improve the NMF algorithms one of the best algorithms remains the classic non-negative quadratic programming solver based on an active set procedure [Lawson and Hanson, 1974; Bro and de Jong, 1997]. This procedure is given in Algorithm 2.4. The method calculates the unconstrained solution of the active set and project this solution optimally back to the positive orthant. Finally, optimization under non-negativity constraints has been well studied within linear and quadratic programming, see for instance [Dantzig, 1963; Saunders, 2003]. Here efficient interior points methods have also been derived imposing non-negativity through log-barrier functions [Rojas and Steihaug, 2002].

Algorithm 2.4: Nonnegative Quadratic Programming $\mathbf{s} = \text{NNQP}(\boldsymbol{H}, \mathbf{f})$ solves $\text{argmin}_{\mathbf{s}} \frac{1}{2}\mathbf{s}^\top \boldsymbol{H}\mathbf{s} - \mathbf{f}^\top \mathbf{s}$ s.t. $0 \leq \mathbf{s}$

$A = \varnothing$, $I = \{1, \ldots, k\}$, $\mathbf{d} = \mathbf{0}$, $\mathbf{w} = \mathbf{f} - \boldsymbol{H}\mathbf{d}$
repeat
  $m = \text{argmax}_{n \in I} w_n$, $A = A \cup \{m\}$, $I = I \setminus \{m\}$
  $\mathbf{s}_A = \boldsymbol{H}_{A,A}^{-1} \mathbf{f}_A$
  while $\min(\mathbf{s}_A) \leq 0$ **do**
    $\beta = -\min_{n \in A} d_n/(d_n - s_n)$
    $\mathbf{d} = \mathbf{d} + \beta(\mathbf{s} - \mathbf{d})$
    $I = I \cup \{n\}$, $A = A \setminus \{n\}$
    $\mathbf{s}_A = \boldsymbol{H}_{A,A}^{-1} \mathbf{f}_A$, $\mathbf{s}_I = \mathbf{0}$
  end while
  $\mathbf{d} = \mathbf{s}$, $\mathbf{w} = \mathbf{f} - \boldsymbol{H}\mathbf{d}$
until $I = \varnothing$ or $\max_{n \in I} w_n \leq 0$

2.5 Sparse Coding

According to Barlow [Barlow, 1961] an important characteristic of sensory processing in the brain is *redundancy reduction*. As Barlow writes in his introduction to the role of redundancy [Barlow, 1996]



Horace Barlow
(1921-)

"What is a pattern? It is some kind of regularity or self-similarity in a signal or set of data. If there is no regularity, or no repetition caused by self-similarity, then surely there is no pattern. But if there is such regularity or repetition, then this is a form of redundancy, and offers the opportunity for recoding to reduce it. Of course the pattern element can be completely arbitrary, a sequence of randomly selected digits for example, but if repeated this element will make a pattern. Thus it seems to me that the importance of redundancy is almost a tautology and follows simply from the nature of pattern."

One method for performing redundancy reduction is Sparse Coding [Field, 1994]. Sparse Coding is a well established principles for feature extraction [Olshausen and Field, 2004, 1996; Hoyer, 2002; Eggert and Körner, 2004; Olshausen and Field, 1997]. The principle of Sparse Coding is to account for as much information as possible while transmitting as little information as possible. Mathematically, this corresponds to attaining as few non-zero elements as possible in the code (i.e., ideally to minimize the l_0-norm). Olshausen and Field [2004] argue that the brain might employ Sparse Coding since it allows for increased storage capacity in associative memories, it makes the structure in natural signals explicit, it represents complex data in a way that is easier to read out at subsequent levels of processing, and it is energy efficient. Thus, sparseness is a natural constraint for Unsupervised Learning and Sparse Coding often results in parsimonious features. As such it was demonstrated in [Olshausen and Field, 1996] how Sparse Coding of image patches resulted in Gabor like features, see also figure 2.5. As Olshausen and Field [1996] writes in their Nature article from 1996

Bruno A.
Olshausen

"We Show that a learning algorithm that attempts to find sparse linear codes for natural scenes will develop a complete family of localized, oriented, bandpass receptive fields, similar to those found in the primary visual cortex. The resulting sparse image code provides a more efficient representation for later stages of processing because it possesses a higher degree of statistical independence among its outputs."

David J. Field

The Sparse Coding model is estimated minimizing an objective function both penalizing reconstruction error as well as deviation from sparsity of the code

$$\text{argmin}_{\boldsymbol{A},\boldsymbol{S}}\, D(\boldsymbol{X}, \boldsymbol{AS}) - \lambda \log sp(\boldsymbol{S}) \tag{2.6}$$

where $D(\cdot,\cdot)$ is a distance measure of the reconstruction error for instance the

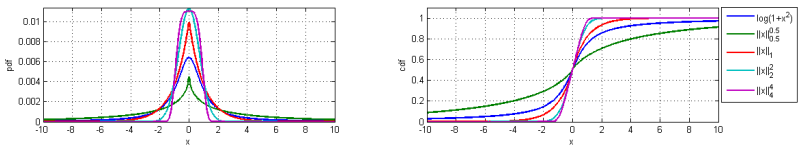


Figure 2.4: Example of probability density functions (pdf) and cumulative distribution functions (cdf) for various choices of sp given in equation (2.6). The corresponding excess kurtosis $kurt_{exc}$ for the priors are $kurt_{exc_{\log(1+x^2)}} = 55.6$, $kurt_{exc_{\|x\|_{0.5}^{0.5}}} = 13.3$, $kurt_{exc_{\|x\|_1}} = 3.1$, $kurt_{exc_{\|x\|_2^2}} = 0.0$ and $kurt_{exc_{\|x\|_4^4}} = -0.8$

least squares error and sp is the sparse prior distribution of $\boldsymbol{S}$. While the optimal degree of sparsity can be stated in terms of the l_0-norm optimizing for the l_0-norm is NP-hard and does not facilitate learning rules based on gradient nor Hessian information. Thus, often sp is given by the laplace distribution $sp(\boldsymbol{S}_{d,j}) \propto e^{-|\boldsymbol{S}_{d,j}|}$ which is the closest convex proxy for the l_0-norm [Donoho, 2006; Mørup et al., 2008d]. As such, Sparse Coding is related to maximum likelihood Independent Component Analysis (ICA) [Bell and Sejnowski, 1995] based on sparse priors. Hence, the Gabor features found by Sparse Coding, see figure 2.5 has also been reported using ICA [Hateren and Van der Schaaf, 1998; Hyvärinen and Hoyer, 2000]. λ is a parameter defining the tradeoff between reconstruction error and sparseness of the code. The probability density function (pdf) and cumulative distribution function (cdf) for various priors (i.e., choices of sp) are given in figure 2.4.

The challenge when solving for the sparse code is to efficiently estimate $\boldsymbol{S}$ for fixed $\boldsymbol{A}$ where $\boldsymbol{A}$ and $\boldsymbol{S}$ are estimated using an alternating approach, i.e. based on a Gauss-Seidel strategy. Normally, the norm of the components of $\mathbf{A}$ is kept fixed either using a Lagrange multiplier approach [Lee et al., 2007] or some projected interior points method [Olshausen and Field, 1996; Blumensath and Davies, 2004a], see also Appendix A.1.3.2. Using least squares as a measure of reconstruction error and the Laplace prior as sparsity distribution the above problem becomes the well known Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996] or Basis Pursuit Denoising (BPD) problem [Shaobing and Donoho, 1994] for fixed $\boldsymbol{A}$

$$\mathbf{LASSO}(\boldsymbol{S}) : \arg\min_{\boldsymbol{S}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \lambda\|\boldsymbol{S}\|_1. \tag{2.7}$$

Although, the above objective is a convex optimization problem no closed form solution exists thus iterative procedures have to be employed in order to solve the problem. Several methods have been proposed: In [Olshausen and Field, 1996] and algorithms based on conjugate gradient (CONJ.GRAD.) was derived. In sparselab (`http://sparselab.stanford.edu/`) the discontinuity of the deriva-

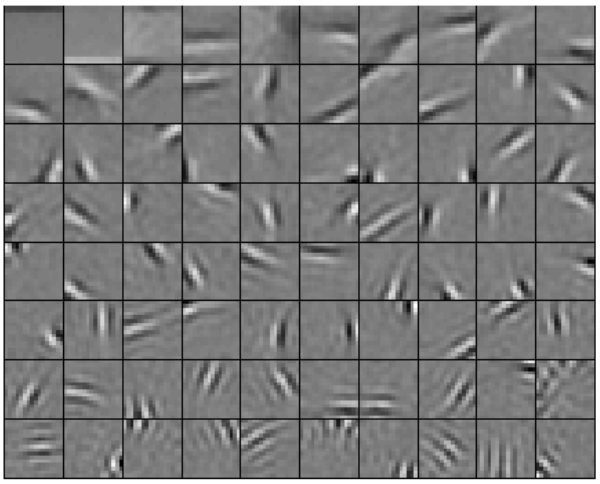


Figure 2.5: Example of feature images of size 16×16 obtained when analyzing the patched natural image data $\boldsymbol{X}^{256\ Pixels \times 10240\ Patches}$ given in [Olshausen and Field, 1996]. Clearly, Gabor like features are obtained as reported in [Olshausen and Field, 1996]. Similar Gabor like features have been reported using ICA [Hateren and Van der Schaaf, 1998; Hyvärinen and Hoyer, 2000].

tive at zero is avoided by turning the problem into a non-negative quadratic programming problem solving for the BPD. In [Lee et al., 2007] the SIGNSEARCH algorithm was formed based on an active set procedure to estimate the sign of $\boldsymbol{S}$ such that a closed form solution could be found as $(\boldsymbol{A}^\top \boldsymbol{A})^{-1}(\boldsymbol{A}^\top \boldsymbol{X} - \lambda \operatorname{sign}(\boldsymbol{S}))$. In [Blumensath and Davies, 2004a] the l_1-penalty was approximated by a quadratic penalty $\|\boldsymbol{S}\|_1 = \sum_{d,j} |\boldsymbol{S}_{d,j}| = \sum_{d,j} \frac{\boldsymbol{S}_{d,j}^2}{\boldsymbol{Q}_{d,j}}$ where $\boldsymbol{Q}_{d,j} = \sqrt{\boldsymbol{S}_{d,j}^2}$ but $\boldsymbol{Q}$ kept fixed when deriving the gradient and Hessian with respect to $\boldsymbol{S}$ despite depending on $\boldsymbol{S}$. This procedure we denote (BD-SC). In [Efron et al., 2004; Osborne et al., 2000] an algorithm was derived to solve for $\boldsymbol{S}$ such that the entire regularization path could be obtained, i.e. solution for all values of λ at the cost of an ordinary least squares solution forming the Least Angle Regression and Selection algorithm (LARS), see also Algorithm 2.5a. This algorithm was extended in [Mørup et al., 2008d] to non-negativity constraints as outlined in Algorithm 2.5b. All the above methods except the conjugate gradient based approach rely on Hessian information thus become very memory intensive for large scale problems. While BPD only require the diagonal Hessian it is still more memory intensive as the problem size is doubled when solved as a non-negative quadratic programming problem.

Algorithm 2.5a: Least Angle Regression and Selection (LARS)

repeat
  $\mathbf{c} = \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$
  $j = \arg\max(|\mathbf{c}_I|)$
  $A = [A\ j]$
  $I = I \setminus j$
  $\boldsymbol{\beta}_A = \boldsymbol{\beta}_A + \mu(\mathbf{X}^\top\mathbf{X})^{-1}_{A,A}\,\mathrm{sign}(\mathbf{c}_A)$
  $\mu = \arg\min_\mu \begin{cases} \exists \widetilde{\boldsymbol{\beta}}_{A_k} = 0 \text{ then } I = [I\ A_k], A = A \setminus A_k \\ \exists l \in I:\ |\widetilde{\mathbf{c}}_l| = |\widetilde{\mathbf{c}}_A| \\ \widetilde{\mathbf{c}}_A = \mathbf{0} \end{cases}$
until $\mathbf{c}_A = \mathbf{0}$

Algorithm 2.5b: Non-neg. Least Angle Regr. and Sel. (NLARS)

repeat
  $\mathbf{c} = \mathbf{X}^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$
  $j = \arg\max(\mathbf{c}_I),\ \mathbf{c}_j > 0$
  $A = [A\ j]$
  $I = I \setminus j$
  $\boldsymbol{\beta}_A = \boldsymbol{\beta}_A + \mu(\mathbf{X}^\top\mathbf{X})^{-1}_{A,A}\mathbf{1}$
  $\mu = \arg\min_\mu \begin{cases} \exists\ \boldsymbol{\beta}_{A_k} = 0 \text{ then } I = [I\ A_k], A = A \setminus A_k \\ \exists l \in I:\ \widetilde{\mathbf{c}}_l = \widetilde{\mathbf{c}}_A \\ \widetilde{\mathbf{c}}_A = \mathbf{0} \end{cases}$
until $\mathbf{c}_A = \mathbf{0}$

Unfortunately, simple gradient based methods normally fails in finding the optimal solution since they tend to get stuck in very small step sizes due to oscillations around zero. To see this consider the gradient of the objective in (2.7) given by

$$\nabla_{\boldsymbol{S}}\mathbf{LASSO}(\boldsymbol{S}) = \boldsymbol{A}^\top(\boldsymbol{AS} - \boldsymbol{X}) + \lambda Sign(\boldsymbol{S}).$$

A gradient based update would then yield

$$\boldsymbol{S}^{new} = \boldsymbol{S}^{old} - \mu(\boldsymbol{A}^\top(\boldsymbol{AS}^{old} - \boldsymbol{X}) + \lambda Sign(\boldsymbol{S}^{old})).$$

However, if $|\lambda Sign(\boldsymbol{S}^{old})_{d,j}| >> |(\boldsymbol{A}^\top(\boldsymbol{AS}^{old} - \boldsymbol{X}))_{d,j}|$ the regularization will dominate the update and rather than forcing $\boldsymbol{S}_{d,j}$ to zero $\boldsymbol{S}_{d,j}$ will cross zero and keep oscillating around zero until the step size μ becomes infinitesimal small despite that the regularization is in fact minimized when elements in $\boldsymbol{S}$ becomes zero, see figure 2.6. At first glance this might appear to be a minor concern, however, when many elements of $\boldsymbol{S}$ are close to zero the joint effect of all these oscillations will completely dominate the update.

To avoid the oscillations, we proposed in [Mørup and Schmidt, 2008] to split the gradient based update into a simple two step procedure; first update the gradient of the reconstruction error and next for the regularization forming the

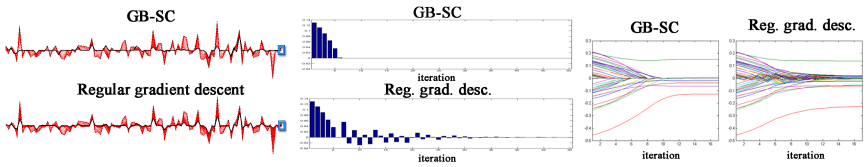


Figure 2.6: Illustration of how the gradient based sparse coding GB-SC alleviates poor convergence due to oscillations around zero. **Left panel:** the progression through 50 iterations of the GB-SC algorithm as well as regular gradient descent on a problem with 100 variables. The dotted black line is the initial solution, the solid black line is the solution obtained after 50 iterations, and the red lines are intermediate results. For the GB-SC, the final solution is the global minimum of the problem. The regular gradient descent algorithm does not converge fast to the optimum, but oscillates around zero. **Middle panel:** Inspection of the progression of one variable (marked by the blue boxes in the left panel). No oscillations around zero are found for the GB-SC based method, whereas the regular gradient descent method oscillates around zero causing the algorithm to suffer from slow convergence. **Right panel:** The progression of the coefficients through 17 iterations for a problem with 50 variables. Even for this relatively small problem, regular gradient descent is stuck in suboptimal solutions due to oscillations around zero, whereas the GB-SC efficiently finds the optimal solutions.

Gradient Based Sparse Coding (GB-SC) algorithm given in Algorithm 2.5c. This simple algorithm avoids the oscillatory behavior encountered in regular gradient descent.

Algorithm 2.5c: Gradient Based Sparse Coding (GB-SC)

repeat
  Update $\boldsymbol{S}$ according to reconstruction penalty
    $\boldsymbol{S}^{new} = \boldsymbol{S} - \mu(\boldsymbol{A}^\top(\boldsymbol{AS} - \boldsymbol{X})$
  Update $\boldsymbol{S}^{new}$ according to the sparsity penalty such that element crossing zero are set to zero
    $$\boldsymbol{S}^{new}_{d,j} = \begin{cases} 0 & \text{if } |\boldsymbol{S}^{new}_{d,j}| < \mu\lambda \\ \boldsymbol{S}^{new}_{d,j} - \mu\lambda Sign(\boldsymbol{S}^{new}_{d,j}) & \text{otherwise} \end{cases}$$
  if LASSO$(\boldsymbol{S}^{new}) <$ **LASSO**$(\boldsymbol{S})$ **then**
    $\mu = 1.2\mu$
    $\boldsymbol{S} = \boldsymbol{S}^{new}$
  else
    $\mu = \mu/2$
  end if
until convergence

	256×100	256×256	256×1000	256×2500
BD-SC	0.3641 ± 0.3044	11.625 ± 4.4922	—	—
SIGNSEARCH	0.0750 ± 0.0359	0.1984 ± 0.1342	$\mathbf{0.3734 \pm 0.1759}$	$\mathbf{1.6969 \pm 0.6441}$
CONJUGATE GRADIENT	0.4172 ± 0.0651	1.1219 ± 0.2560	9.0297 ± 1.8055	45.6297 ± 12.0142
LARS	0.0453 ± 0.0226	$\mathbf{0.1313 \pm 0.0787}$	0.4313 ± 0.1477	1.9813 ± 0.6342
BPD	0.5703 ± 0.0696	0.9313 ± 0.0748	2.8719 ± 0.1389	15.5047 ± 0.7882
GB-SC	$\mathbf{0.0125 \pm 0.0066}$	0.3172 ± 0.2121	2.0688 ± 1.0760	22.8828 ± 12.2846

Table 2.1: Comparison of the CPU time for various Sparse Coding algorithms on different problem sizes. The problem solved is $\arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \boldsymbol{A}\mathbf{s}\|_F^2 + \lambda\|\mathbf{s}\|_1$, for $\lambda = 0.05$. $J \times D$ denotes the size of $\boldsymbol{A}$, (J image pixels and D basis vectors.) The mean and standard deviation is given for 10 randomly generated problems, each given by setting $\boldsymbol{A}$ to D randomly chosen image patches from the natural images data set [Olshausen and Field, 1996] and $\mathbf{x}$ to a randomly selected image patch, not already used in the dictionary, $\boldsymbol{A}$. Notice, BD-SC, SIGNSEARCH and LARS all find the global optimum. The remaining algorithms were stopped, when their deviation from the true minimum was less than 10^{-4}. For $D \leq J$, the GB-SC was the fastest of all the algorithms, but for over-complete problems, i.e., $D \gg J$, the GB-SC algorithm is not in general as effective as the other algorithms, which use Hessian information; however, it is still faster than the conjugate gradient based method. Hence, the GB-SC algorithm is not only simple, but also efficient, and even outperforms state of the art algorithms for $D \leq J$. BD-SC for 256×1000 and 256×2500 was not included, as it was more than 100 times slower than the conjugate gradient algorithm. The conjugate gradient algorithm was obtained from `www.l1-magic.org`, whereas the BPD and LARS were obtained from `www.sparselab.stanford.edu`. The SIGNSEARCH algorithm was kindly provided by H. Lee [Lee et al., 2007].

Notice, although the procedure in algorithm 2.5c is given for least squares minimization with sparsity penalty based on the Laplace prior, i.e. the l_1-norm, the approach of splitting the gradient into an update for the reconstruction error and an update for the sparsity penalty generalizes to other types of reconstruction metrics and sparsity penalty measures. Table 2.5 demonstrates the performance of the afore mentioned algorithms to solve for the sparse codes.

What is the "correct" degree of sparseness is an open problem. In [Hoyer, 2004] the sparseness of the decomposition was controlled by the sparseness measure

$$sp_{\text{PH}-\text{L1/L2}}(\boldsymbol{S}_{d,:}) = \frac{1}{\sqrt{J}-1}(\sqrt{J} - \frac{\|\boldsymbol{S}_{d,:}\|_1}{\|\boldsymbol{S}_{d,:}\|_2}), \tag{2.8}$$

where $\boldsymbol{S}_{d,:}$ denotes the d^{th} row of $\boldsymbol{S}$. Notice, $sp_{\text{PH}-\text{L1/L2}}$ takes values between 0 and 1 based on the NMFSC algorithm described in [Hoyer, 2004]. Solving for a specific sparsity level for each component is a difficult problem. However, solving for a specific sparsity on the full matrix $\boldsymbol{S}$ mounts to controlling the single parameter λ which we demonstrated in [Mørup et al., 2008d] has a simple solution. Through the LARS and NLARS the sparsity can be directly controlled

for the following two types of sparsity measures on the full matrix $\boldsymbol{S}$

$$sp_{\mathrm{L0}}(\boldsymbol{S}) = 1 - \frac{\|\boldsymbol{S}\|_0}{DJ} \tag{2.9}$$

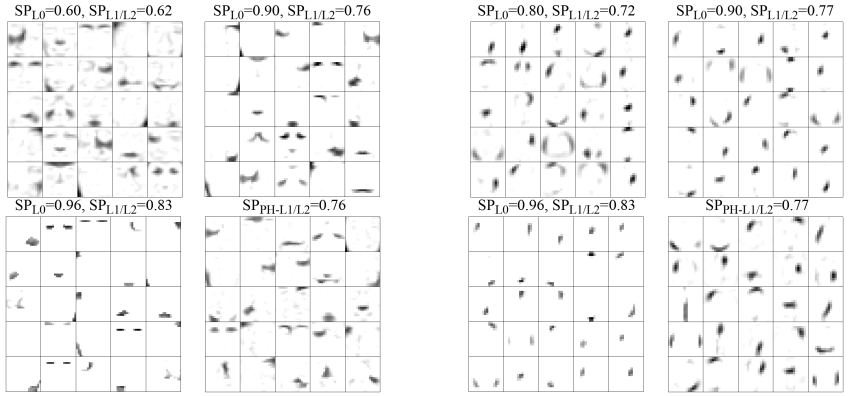
$$sp_{\mathrm{L1/L2}}(\boldsymbol{S}) = \frac{1}{\sqrt{DJ}-1}(\sqrt{DJ} - \frac{\|\boldsymbol{S}\|_1}{\|\boldsymbol{S}\|_2}). \tag{2.10}$$

The measure $sp_{\mathrm{L1/L2}}(\mathbf{S})$ is equivalent to the measure $sp_{\mathrm{PH-L1/L2}}(\boldsymbol{S}_{d,:})$, but for the full matrix rather than for each row of $\boldsymbol{S}$. On the other hand sp_{L0} gives direct control of the percentage of active elements in $\boldsymbol{S}$, i.e., the L_0 norm of $\boldsymbol{S}$. Both measures take values between 0 and 1, for instance $sp_{\mathrm{L0}}(\boldsymbol{S}) = 1$ implies that all elements of $\boldsymbol{S}$ are turned off while $sp_{\mathrm{L0}}(\boldsymbol{S}) = 0$ that all elements are nonzero. The effect of imposing sparseness to various degree according to the above measures can be seen in Figure 2.7.

While the above measures quantifies the degree of sparseness, the optimal degree of sparsity is still an open problem. We return to the quote of Thurstone: *"In a factor problem one is concerned about how to account for the observed correlations among all the variables in terms of the smallest number of factors and with the smallest possible residual error"*. Thus, the "correct" degree of sparseness must be equivalent to the degree of sparsity for which the smallest number of active variables are attained such that the residual error is unaffected. Consequently, we proposed in [Mørup et al., 2008e] to use an L-curve approach [Hansen, 1992] to select for the sparsity degree λ plotting the reconstruction error vs. the L0-norm, i.e. inspecting the plot $(\|\boldsymbol{S}\|_0, D(\cdot,\cdot))$. The region of maximum curvature will give a relative indication of where the quality of the reconstruction error begins to be affected when increasing the regularization strength λ. Thus, a value of λ just after this corner region is where the regularization keeps decreasing the number of variables without severely affecting the reconstruction error, see also Figure 2.8.

2.6 Other Types of Matrix Decompositions

The above constraints to the factor analysis model given by imposing independence, non-negativity and sparseness can be combined to form even more restricted models, see for instance [Plumbley, 2003]. Furthermore, objectives based on other types of metrics than least squares for the quality of reconstruction are conceivable. For instance, the l_p-norm, Bregman, Csizcar and alpha divergences [Blumensath and Davies, 2004a; Dhillon and Sra, 2006; Cichocki et al., 2006] might be more relevant as measures of deviation between data and representation when the noise can not be assumed Gaussian. It is worth noting that an algorithm similar to LARS has been derived forming the full-regularization



(a) MIT face data (b) USPS handwritten digits

Figure 2.7: **Left panel:** Result obtained analyzing the CBCL face database using the NLARS to control the sparsity of the decomposition. Top left panel: Sparsity set to $sp_{\mathrm{L0}} = 0.6$. Top right panel: Sparsity set to $sp_{\mathrm{L0}} = 0.9$. Bottom left panel: Sparsity set to $sp_{\mathrm{L0}} = 0.96$. Bottom right panel: Sparsity set to $sp_{\mathrm{PH-L1/L2}} = 0.76$ corresponding to the value of sparsity given in the top right panel of $sp_{\mathrm{L1/L2}} = 0.76$. The result obtained by the proposed NLARS resembles well the results obtained by the NMFSC algorithm. However, regularizing on the full matrix $\boldsymbol{S}$ does not fix the degree of sparsity within each component and as such each component has different degree of sparsity. Thus, component 16 is completely turned off when $sp_{\mathrm{LO}} = 0.96$. **Right panel:** Result obtained analyzing the USPS handwritten digit database using the NLARS to control the sparsity of the decomposition. Top left panel: Sparsity set to $sp_{\mathrm{L0}} = 0.8$. Top right panel: Sparsity set to $sp_{\mathrm{L0}} = 0.9$. Bottom left panel: Sparsity set to $sp_{\mathrm{L0}} = 0.96$. Bottom right panel: The NMFSC algorithm with sparsity set to $sp_{\mathrm{PH-L1/L2}} = 0.77$ corresponding to the value of sparsity given in the top right panel of $sp_{\mathrm{L1/L2}} = 0.77$. Again the solution for $sp_{\mathrm{L0}} = 0.9$ having $sp_{\mathrm{L1/L2}} = 0.77$ correspond well to the solution obtained for $sp_{\mathrm{PH-L1/L2}} = 0.77$ using the NMFSC algorithm.

path for Least Absolute Deviation (LAD) (i.e., l_1-norm minimization) [Wang et al., 2006; Wu and Lange, 2008] while an algorithmic framework to approximately find the full-regularization path for objective functions in general has been proposed in [Rosset, 2005].

A multitude of matrix decomposition models other than the above have been proposed. One of many is the sparse PCA model proposed in [Zou et al., 2006] given by

$$\boldsymbol{X} \approx \boldsymbol{AWX}$$

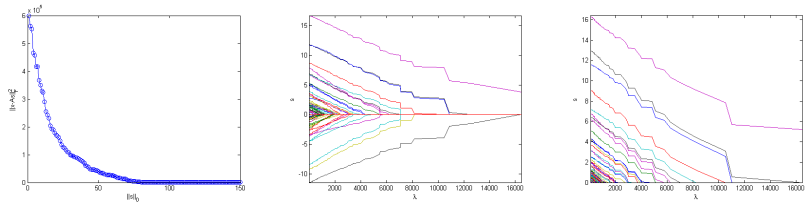


Figure 2.8: **Left panel:** Evaluation of the tradeoff between reconstruction and sparsity of the code given by a plot of the reconstruction error vs. the l_0-norm of the code. A candidate for a good value of sparsity imposed is the region where the inclusion of more variables does not significantly improve the reconstruction error. This corresponds to the regularization value just after the region with maximal curvature on the curve. **Middle panel:** Example of the full regularization path obtained using the LARS algorithm. **Right panel:** Example of the full regularization path for the same problem obtained using the NLARS algorithm such that all coefficients are constrained positive. Notice, the piece-wise linearity forming the full regularization paths for both the LARS and NLARS. The paths are shown from the point where the second parameter enter the active set.

such that $\boldsymbol{W}$ is sparse. Thus, the desire here is to describe the data as a linear combination of bases that are a combination of a few of the original data points. Thus, the model corresponds to the regular Principal Component Analysis with $\boldsymbol{S} = \boldsymbol{W}\boldsymbol{X}$. The model can be estimated solving the following optimization problem

$$\begin{array}{ll} \text{argmin}_{\boldsymbol{A},\boldsymbol{W}} & \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{W}\boldsymbol{X}\|_F^2 + \lambda_1\|\boldsymbol{W}\|_1 + \lambda_2\|\boldsymbol{W}\|_F^2 \\ \text{subjected to} & \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I} \end{array}$$

Where λ_1 and λ_2 are regularization parameters for the l_1 and l_2 norm respectively.

Yet another type of decomposition is the CUR decomposition (see for instance [Mahoney et al., 2006]) given by

$$\boldsymbol{X}^{I\times J} \approx \boldsymbol{C}^{I\times D'}\boldsymbol{U}^{D'\times D}\boldsymbol{R}^{D\times J}$$

Where $\boldsymbol{C}$ and $\boldsymbol{R}$ are given columns and row of the original data and $\boldsymbol{U}$ relates the estimated row and columns spaces. Hence, the decomposition represents the data in terms of original row and column observation which facilitates component interpretation, thus, $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{T}$, $\boldsymbol{R} = \boldsymbol{W}\boldsymbol{X}$ where $\boldsymbol{T}$ and $\boldsymbol{W}$ are binary matrices selecting rows and columns of $\boldsymbol{X}$. The CUR decomposition can therefore be considered a strongly restricted sparse PCA where $\boldsymbol{W}$ is binary with only one non-zero entry in each row while $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{T}\boldsymbol{U}$. Thus, also the sparse

PCA and CUR model attempt to describe the data through a simplistic representation.

2.7 Model Selection

Estimating the adequate number of components D to model the data is an open problem. For the SVD the eigenvalue-spectrum has often been evaluated and a threshold chosen such that eigenvectors pertaining to eigenvalues less than this threshold are considered to not significantly contribute to the description of the data.

Other approaches for estimating D include various types of information criteria. Let L be the likelihood of the model (see also Appendix A.2), n the number of observations and k the number of parameters in the model. Akaike [1974] derived in 1974 the Akaikes Information Criterion (AIC) given by

$$AIC = -\log L + k$$

Inspired by this criterion Schwarz [1978] derived in 1978 the Bayesian Information Criterion (BIC) given by

$$BIC = -\log L + k\frac{\log n}{2}$$

Both criteria evaluate a tradeoff between goodness of fit (the negative log-likelihood $-\log L$) and complexity of the model (k, $k\frac{\log n}{2}$ respectively) such that the minimal value of these criteria indicates the model which is the most adequate. Notice how BIC tends to penalize model complexity more heavily than AIC thus giving a more conservative estimate of what is considered the best model. When two models equally well describe the data the criteria above will favor the simpler of the models. As Schwarz [1978] writes

Gideon Schwartz (1933-2007)

"Qualitatively both our procedure and Akaike's give a mathematical formulation of the principle of parsimony in model building"

Hirotsugu Akaike (1927-)

In [Hansen and Larsen, 1996; Hansen et al., 1999] a similar criterion to AIC was derived and the number of components estimated by splitting the data into a training set used to find model parameters and a test set used to estimate the generalization error. We used a similar approach in [Rasmussen et al., 2008] to estimate the number of components in an ICA model. Finally, the evidence (i.e.

marginal likelihood, see Appendix A.2) of the model have been approximated by more involved methods such as variational Bayes or sampling-based approaches [Beal, 2003]. As pointed out by Beal [2003] evaluating the evidence relates to the principle of parsimony

> *"Integrating out the parameters* (required when calculating the evidence) *penalises models with more degrees of freedom since these models can a priori model a larger range of data sets."*

2.8 Summary

For all the above models the principle of parsimony, i.e., finding the representation that most simply accounts for the data has been the main goal. As previously described there is a direct link between kurtosis maximization in ICA and Quartimax/Varimax rotation. Furthermore, maximum likelihood ICA based on sparse priors is strongly related to Sparse Coding such that the prior in ML ICA relates to the measure of deviation from sparsity in Sparse Coding. However, while the rotational approaches account for the maximal amount of variance in the data, Sparse Coding introduces a bias when penalizing deviation from sparsity. The part based representation of NMF also attain sparse solutions since there is no cancellation between components thus the whole is described by its constituting parts. As such, all the above approaches are strongly related to redundancy reduction as coined by Barlow. Within neuroscience, the principle of redundancy reduction explains aspects of the early processing of sensory data by the brain. Thus, the behavior of the aforementioned models also relate to neural information processing. Thus, parsimony, redundancy reduction, sparsity of the code, part based representation and independence are all interrelated goals that attempt to account for the data in a simplistic way. Finally, simplicity was also the goal of the various model selection criteria from AIC and BIC to the more involved approaches evaluating the evidence of the data to avoid overfitting. Thus, we finish this chapter by looking at the insightful William of Ockham.

William of Ockham (1288-1347)

Occam's razor is named after William of Ockham who stated that the explanation of any phenomenon should make as few assumptions as possible, eliminating those that make no difference in the observable predictions of the explanatory hypothesis or theory. The principle is in Latin expressed as "lex parsimoniae" that is "law of parsimony" or "law of succinctness". This is often paraphrased as "All other things being equal, the simplest solution is the best." *(source: wikipedia.org)*

Chapter 3

Tensor Decomposition

Tensor decompositions are in frequent use today in a variety of fields including psychometric, chemometrics, image analysis, graph analysis and signal processing [Murakami and Kroonenberg, 2003; Vasilescu and Terzopoulos, 2002; Wang and Ahuja, 2003; Jia and Gong, 2005; Sun et al., 2005; Gurden et al., 2001; Nørgaard and Ridder, 1994; Smilde et al., 1999, 2004; Andersson and Bro, 1998]. Tensors, i.e., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, also called multi-way arrays or multidimensional matrices are generalizations of vectors (first order tensors) and matrices (second order tensors). Tensor decompositions originated with Hitchcock in 1927 [Hitchcock, 1927b,a]. However, the idea of a multi-way model is attributed to Cattell in 1944 [Cattell, 1944, 1952]. Nevertheless, these concepts received little attention until the work of Tucker in the 1960s [Tucker, 1966] and Carroll and Chang [Carroll and Chang, 1970] and Harshman [Harshman, 1970] in 1970.

A recent, thorough review paper on tensor decomposition can be found in [Kolda and Bader, 2008]. Other good introductions to the field are [Bro, 1998; Smilde et al., 2004]. For aspects on computation see also [Tomasi, 2006]. The present description only emphasizes some of the most important models and aspects of tensor decomposition.

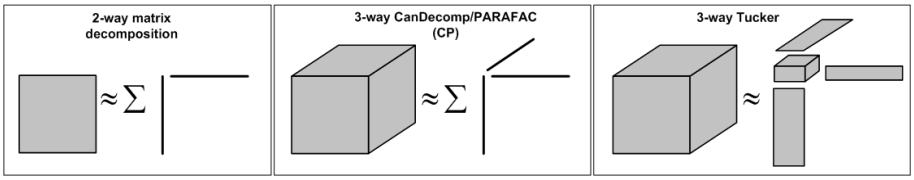


Figure 3.1: The CP and Tucker model both generalizes the 2-way matrix decomposition to multi-way arrays. **Left panel:** The factor analysis based decomposition decompose the data into a sum of rank 1 components. **Middle panel:** The CP model generalizes the sum of rank 1 components straight forward to more modalities by introducing vectors pertaining to these extra modalities. **Right panel:** The Tucker model also decompose the data into vectors pertaining to each modality but through a core array the components of each modality can interact across the modalities regardless of their component index. While the CP model is closely related to the rank of a tensor, the Tucker model generalizes compression by orthogonal matrices to tensors.

3.1 The CandeComp/PARAFAC model

The CandeComp/PARAFAC (CP) model independently proposed by [Hitchcock, 1927b; Carroll and Chang, 1970; Harshman, 1970] reads

$$x_{i_1 i_2 \ldots i_N} \approx \sum_d \prod_{n=1}^{N} a^{(n)}_{i_n,d},$$

i.e., the data is decomposed into a sum of rank one components. Therefore, the minimal D for which the approximation is exact yield the rank of the tensor $\mathcal{X}$. For the k^{th} slab, $\boldsymbol{X}_{:,:,k}$, the CP model for a 3-way array reads

$$\begin{aligned} \boldsymbol{X}_{(:,:,k)} \approx \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{B}^T &= (\boldsymbol{A}\boldsymbol{T})(\boldsymbol{T}^{-1}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{Q})(\boldsymbol{Q}^{-1}\boldsymbol{B}^T) \\ &= \widehat{\boldsymbol{A}}\,\mathrm{diag}(\widehat{\boldsymbol{C}}_{k,:})\widehat{\boldsymbol{B}}^T. \end{aligned}$$

Thus, if two solutions $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}, \widehat{\boldsymbol{C}}$ exists there must be a mapping from one solution to the other given by $\boldsymbol{T}$ and $\boldsymbol{Q}$. However, for this mapping the term $\boldsymbol{T}^{-1} diag(\boldsymbol{C}_{k,:})\boldsymbol{Q}$ has to be diagonal for all k which when $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ have full rank restricts $\boldsymbol{T}$ and $\boldsymbol{Q}$ to be simple scale and permutation matrices [Harshman, 1972a; Möcks, 1988]. In [Kruskal, 1976, 1977] the uniqueness properties of the CP model were thoroughly investigated and the following uniqueness criterion generalized to N-ways arrays in [Sidiropoulos and Bro, 2000] given

$$\sum_{n=1}^{N} k_{\boldsymbol{A}^{(n)}} \geq 2D + N - 1. \tag{3.1}$$

The Kruskal rank or k-rank $k_{\boldsymbol{A}}$ of a matrix $\boldsymbol{A}$ is the maximal number r such that any set of r columns of the matrix $\boldsymbol{A}$ is linearly independent. As Kruskal wrote in [Kruskal, 1976], struck by his own uniqueness criterion,

Joseph B. Kruskal
1928-

"A surprising fact is that the nonrotatability characteristic can hold even when the number of factors extracted is greater than every dimension of the three-way array."

Thus, contrary to the regular factor analysis model, the CP model is attractive since it will, under very mild conditions, admit a unique internal representation of the data.

Unfortunately, degenerate solutions are known to occur. I.e., solutions in which the component loadings are highly correlated in all the modes. This makes the CP estimation unstable, slow in convergence and difficult to interpret since the components are dominated by strong cancellation effects between the various components in the model [Harshman and Lundy, 1984], see also Figure 3.3. Furthermore, the decomposition does not admit any known closed form solution as is the case in terms of the SVD for 2-way matrix decomposition, thus, there is in general no guarantee the optimal solution can be attained.

3.1.1 Model estimation

Through the matrizicing operation, $\boldsymbol{X}_{(n)}$, and Khatri-Rao product, $\odot$, the model can be written as a regular factor analysis based decomposition

$$\boldsymbol{X}_{(n)} \approx \boldsymbol{A}^{(n)}\boldsymbol{Z}, \text{ where } \boldsymbol{Z}^\top = \boldsymbol{A}^{(\boldsymbol{N})} \odot \boldsymbol{A}^{(\boldsymbol{N-1})} \odot \ldots \odot \boldsymbol{A}^{(\boldsymbol{n+1})} \odot \boldsymbol{A}^{(\boldsymbol{n-1})} \odot \ldots \odot \boldsymbol{A}^{(\boldsymbol{1})}$$

For the least squares objective for a three way array we get

$$\begin{aligned}
\boldsymbol{X}_{(1)} \approx \boldsymbol{A}^{(1)}(\boldsymbol{A}^{(\boldsymbol{3})} \odot \boldsymbol{A}^{(\boldsymbol{2})})^\top \quad &\Rightarrow \quad \boldsymbol{A}^{(1)} \leftarrow \boldsymbol{X}_{(1)}(\boldsymbol{A}^{(\boldsymbol{3})} \odot \boldsymbol{A}^{(\boldsymbol{2})})(\boldsymbol{A}^{(\boldsymbol{3})^\top}\boldsymbol{A}^{(\boldsymbol{3})} \bullet \boldsymbol{A}^{(2)^\top}\boldsymbol{A}^{(2)})^{-1} \\
\boldsymbol{X}_{(2)} \approx \boldsymbol{A}^{(\boldsymbol{2})}(\boldsymbol{A}^{(\boldsymbol{3})} \odot \boldsymbol{A}^{(\boldsymbol{1})})^\top \quad &\Rightarrow \quad \boldsymbol{A}^{(\boldsymbol{2})} \leftarrow \boldsymbol{X}_{(2)}(\boldsymbol{A}^{(\boldsymbol{3})} \odot \boldsymbol{A}^{(\boldsymbol{1})})(\boldsymbol{A}^{(\boldsymbol{3})^\top}\boldsymbol{A}^{(\boldsymbol{3})} \bullet \boldsymbol{A}^{(1)^\top}\boldsymbol{A}^{(1)})^{-1} \\
\boldsymbol{X}_{(3)} \approx \boldsymbol{A}^{(\boldsymbol{3})}(\boldsymbol{A}^{(\boldsymbol{2})} \odot \boldsymbol{A}^{(\boldsymbol{1})})^\top \quad &\Rightarrow \quad \boldsymbol{A}^{(\boldsymbol{3})} \leftarrow \boldsymbol{X}_{(3)}(\boldsymbol{A}^{(\boldsymbol{2})} \odot \boldsymbol{A}^{(\boldsymbol{1})})(\boldsymbol{A}^{(\boldsymbol{2})^\top}\boldsymbol{A}^{(\boldsymbol{2})} \bullet \boldsymbol{A}^{(1)^\top}\boldsymbol{A}^{(1)})^{-1}
\end{aligned}$$

However, forming the Khatri-Rao product is expensive. Thus, the following approach based on pre-multiplying the largest mode(s) with the data is much more computationally efficient [Tomasi, 2006]. Multiplying the first mode with

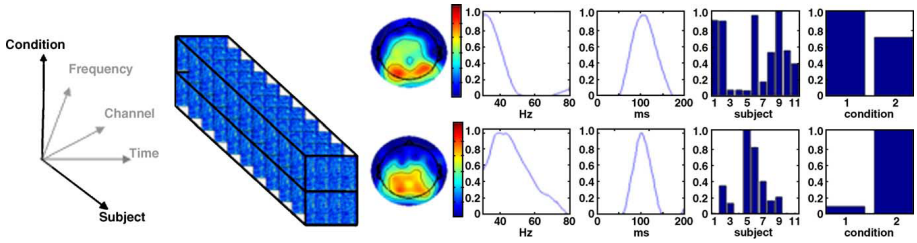


Figure 3.2: Example of a 5-way CP analysis of the inter trial phase coherence of event related EEG data, for details see [Mørup et al., 2006a]. The analyzed data comprise the inter-trial phase coherence (ITPC) recorded through several electrodes for given frequencies at given time points across several subjects through two conditions one where a visual stimuli of objects of known shapes are presented the other of unknown shapes. Thus, the data has the following 5-way structure $channel \times frequency \times time \times subject \times condition$ given to the left. To the right is given the two extracted CP components. Whereas the first component indicates a quantitative difference between the ITPC in the occipital region across the two conditions, the second component indicates a qualitative difference where a slightly more anterior higher frequency activity prevails in the non-object condition. Thus, the CP model can summarize effects that are otherwise difficult to grasp from the raw data.

the data when updating for the second and third mode of a 3-way array gives

$$\begin{aligned}
\boldsymbol{A}^{(1)} &\leftarrow \boldsymbol{X}_{(1)}(\boldsymbol{A}^{(3)} \odot \boldsymbol{A}^{(2)})(\boldsymbol{A}^{(3)\top}\boldsymbol{A}^{(3)} \bullet \boldsymbol{A}^{(2)\top}\boldsymbol{A}^{(2)})^{-1} \\
\widehat{\boldsymbol{X}}_{(1)} &= \boldsymbol{A}^{(1)\top}\boldsymbol{X}_{(1)} \\
\boldsymbol{A}^{(2)} &\leftarrow \widehat{\boldsymbol{X}}_{(2)}(\boldsymbol{A}^{(3)} \odot \boldsymbol{I})(\boldsymbol{A}^{(3)\top}\boldsymbol{A}^{(3)} \bullet \boldsymbol{A}^{(1)\top}\boldsymbol{A}^{(1)})^{-1} \\
\boldsymbol{A}^{(3)} &\leftarrow \widehat{\boldsymbol{X}}_{(3)}(\boldsymbol{A}^{(2)} \odot \boldsymbol{I})(\boldsymbol{A}^{(2)\top}\boldsymbol{A}^{(2)} \bullet \boldsymbol{A}^{(1)\top}\boldsymbol{A}^{(1)})^{-1}
\end{aligned}$$

3.2 The Tucker Model

The Tucker model independently proposed in [Hitchcock, 1927a; Tucker, 1966] reads

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \sum_{j_1 j_2 \cdots j_N} \mathcal{G}_{j_1,j_2,\ldots,j_N} \boldsymbol{A}^{(1)}_{i_1,j_1} \boldsymbol{A}^{(2)}_{i_2,j_2} \cdot \ldots \cdot \boldsymbol{A}^{(N)}_{i_N,j_N},$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$ and $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$. To indicate how many vectors pertain to each modality it is customary also to denote the model a Tucker

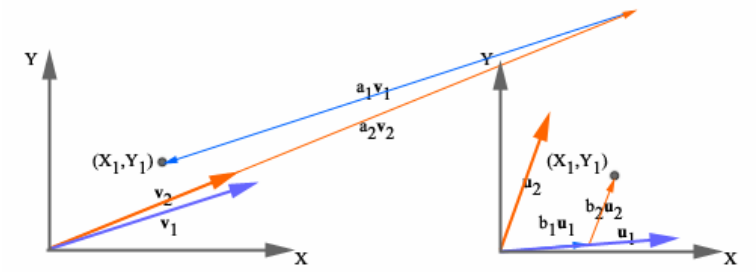


Figure 3.3: Illustration of degeneracy in the CP model. When the solution degenerate the loadings of each mode becomes highly collinear. Thus, despite the vectors to the left and right both span all points in 2D space the collinearity of the vectors to the left causes a strong degree of cancellation among the components hampering interpretability of the decomposition. However, when imposing non-negativity, cancellation can no longer occur thus the decomposition can no longer degenerate as demonstrated by the non-negative span to the right.

$J_1 - J_2 - \cdots - J_N$. Using the n-mode tensor product $\times_n$ [Lathauwer et al., 2000; Tucker, 1966] the model can be written as

$$\mathcal{X} \approx \mathcal{G} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 ... \times_N \boldsymbol{A}^{(N)}.$$

The Tucker model represents the data spanning the n^{th} modality by the vectors (loadings) given by the J_n columns of $\boldsymbol{A}^{(n)}$ such that the vectors of each modality interact with the vectors of all remaining modalities with strengths given by a so-called core tensor $\mathcal{G}$. As a result, the Tucker model encompass all possible linear interactions between vectors pertaining to the various modalities of the data. The CP model is a special case of the Tucker model where the size of each modality of the core array $\mathcal{G}$ is the same, i.e., $J_1 = J_2 = \cdots = J_N$ while interaction is only between columns of same indices such that the only non-zero elements are found along the diagonal of the core, i.e., $\mathcal{G}_{j_1,j_2,...,j_N} \neq 0$ iff $j_1 = j_2 = ... = j_N$. Notice, in the Tucker model a rotation of a given loading matrix $\boldsymbol{A}^{(n)}$ can be compensated by a counter rotation of the core $\mathcal{G}$, i.e., $\mathcal{G} \times_n \boldsymbol{A}^{(n)} = (\mathcal{G} \times_n \boldsymbol{P}^{-1}) \times_n (\boldsymbol{A}^{(n)}\boldsymbol{P})$. Thus, contrary to the CP model, the Tucker model is not generally unique. While the factors of the unconstrained Tucker model can be constrained orthogonal (which is useful for compression) without hampering the reconstruction error, this is not the case for the factors of the CP model.

3.2.1 Model estimation

Traditionally, the Tucker model has been estimated using various alternating least squares algorithms where the columns of $\mathbf{A}^{(n)}$ for the unconstrained Tucker are orthogonal [Andersson and Bro, 1998]. An algorithm generalizing the SVD to the Tucker model based on solving N eigenvalue problems has also been derived forming the higher order singular value decomposition (HOSVD)[Lathauwer et al., 2000].

Using the n-mode matricizing and kronecker product operation the Tucker model can be written as

$$\boldsymbol{X}_{(n)} \approx \boldsymbol{A}^{(n)}\boldsymbol{Z}^{(n)} \quad where \quad \boldsymbol{Z}_{(n)} = \mathbf{G}_{(n)}(\mathbf{A}^{(N)} \otimes ... \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes ... \otimes \mathbf{A}^{(1)})$$

Thus, the model reduces to a sequence of regular factor analysis problem. As a result, for least squares minimization this can be solved by pseudo-inverses, i.e. $\boldsymbol{A}^{(n)} \leftarrow \boldsymbol{X}_{(n)}\boldsymbol{Z}^{(n)^{\dagger}}$. Again using pseudo-inverses the core can be estimated as

$$\mathcal{G} \leftarrow \mathcal{X} \times_1 \boldsymbol{A}^{(1)^{\dagger}} \times_2 \boldsymbol{A}^{(2)^{\dagger}} \times_3 ... \times_N \boldsymbol{A}^{(N)^{\dagger}}.$$

3.3 Core Consistency Diagnostic

While the CP is given by a sum of rank one components, i.e. the straight forward generalization of matrix rank to tensors, the Tucker model is strongly related to data compression through the orthogonal projections compressing the data to the core array $\mathcal{G}$ and can through the HOSVD be considered a generalization of the SVD to tensors. Since the CP model is a special case of the Tucker model where the core array $\mathcal{G} = \mathcal{I}$, i.e., is diagonal with ones across the diagonal, the Tucker model can be used to evaluate the cross-talk between components of the CP model. A measure of this is the core consistency diagnostic proposed in [Bro and Kiers, 2003]

$$CorConDiag = 100 \cdot (1 - \frac{\sum_{i_1,i_2,...,i_n}(\mathcal{G}_{i_1,i_2,...,i_n} - \mathcal{I}_{i_1,i_2,...,i_n})^2}{\sum_{i_1,i_2,...,i_n}\mathcal{I}_{i_1,i_2,...,i_n}})$$

Where $\mathcal{G}$ is estimated as

$$\mathcal{G} \leftarrow \mathcal{X} \times_1 \boldsymbol{A}_{CP}^{(1)^{\dagger}} \times_2 \boldsymbol{A}_{CP}^{(2)^{\dagger}} \times_3 ... \times_N \boldsymbol{A}_{CP}^{(N)^{\dagger}}.$$

where $\boldsymbol{A}_{CP}^{(n)}$ is the n-mode loadings of the CP solution. This $CorConDiag$ has been used to estimate the adequate number of components, D, in the CP model. Too many components will result in a strong degree of cross talk across

the loadings of the modes thus will yield a low value of the $CorConDiag$. Too few components will not have any cross-talk at all. Thus, the "correct" number of components is taken to be just before a major drop-off in the curve of $\{d, CorConDiag\}$. As Bro and Kiers [2003] explain

> *"As a rule of thumb, a core consistency above 90% can be interpreted as 'very trilinear', whereas a core consistency in the neighborhood of 50% would mean a problematic model with signs of both trilinear variation and variation which is not trilinear. A core consistency close to zero or even negative implies an invalid model, because the space covered by the component matrices is then not primarily describing trilinear variation."*

Thus, the Core Consistency Diagnostic can be considered a measure of deviation from trilinearity, i.e., a measure of simplicity of the core.

3.4 Constrained Tensor Decomposition

3.4.1 Orthomax rotation

Contrary to the CP model the Tucker model is not unique, thus, as for the factor analysis type decomposition additional constraints are required to enforce uniqueness of the decomposition. One choice of transformation is to simplify the core structure in some way so that most of the elements of $\mathcal{G}$ are zero, thereby eliminating interactions between corresponding components (diagonalizing the core is in general impossible). This was first observed by Tucker [Tucker, 1966] and in [Kiers, 1998] an Orthomax type of function on the core was optimized to simplify the core.

3.4.2 Independent CP and Tucker

For independent CP, independence is imposed on one of the modalities. Traditionally, orthogonality constraints have been imposed on one of the modes of the CP model to alleviate degeneracy [Möcks, 1988; Field and Graupe, 1991], however, in [Beckmann and Smith, 2005] a tensor ICA method was proposed based on the CP model. Here, independence was simply imposed by matricizing the data to solve a regular ICA problem on the matricized array. However, in [De Vos M., 2007] it has been argued that this is not a "correct" generalization of independence to tensors. Instead a method for independent CP was proposed

that explicitly takes into account the multi-way structure of the data when imposing independence [De Vos M., 2007]. Similar to the approach of Beckman for independence in the CP, independence has been imposed on the Tucker model by matricizing the tensor and applying regular ICA on the matricized array [Vasilescu and Terzopoulos, 2005]. However, this approach, again, does not take into account the multi-way structure of the data when imposing independence.

3.4.3 Non-negative Tensor Factorization

Fully Non-negative Tensor Factorization (NTF) based on the CP model was proposed in [Krijnen and Berge, 1992; Paatero, 1997; Reis et al., 2001] and optimized using multiplicative updates in [Welling and Weber, 2001] naming the decomposition Positive Tensor Factorization (PTF). Much focus has been given to this type of decomposition lately as, contrary to the unconstrained CP model, NTF cannot degenerate [Lim and Golub, 2006], see also figure 3.3 while the decomposition as for NMF yield easily interpreted part based representations [Mørup et al., 2008b]. A fully non-negative Tucker decomposition was proposed in [Mørup et al., 2008b] in combination with sparseness constraints to solve for the non-uniqueness of the decomposition forming the sparse non-negative Tucker (SN-TUCKER), see Figure 3.4. In [Mørup et al., 2006a; Griskova et al., 2007; Arnfred et al., 2007; Mørup et al., 2008b; Arnfred et al., 2008] we demonstrated the usefulness of these types of decompositions when analyzing wavelet transformed event related EEG data. We further developed the toolbox `www.erpwavelab.org` described in [Mørup et al., 2007a] for this purpose, see Figure 3.5.

3.4.4 Sparse CP and Tucker

An inherent problem for non-negative CP is that there is no guarantee of uniqueness of the decomposition, i.e. several positive rank (p-rank) decompositions exists for a tensor with p-rank r [Lim and Golub, 2006]. Thus, to alleviate this uniqueness issue for the CP model imposing sparseness is useful. As the Tucker model is not unique sparseness can improve the uniqueness of the decomposition as well as its interpretability along the lines of the Orthomax rotation of the core array. Finally, regularization in the CP model is known to alleviate degeneracy [Paatero, 2000]. Thus, we proposed in [Mørup et al., 2008b] to impose sparseness in the Tucker model and as such also in the CP model as a special case. We further demonstrated how imposing sparseness on the core can help select which model is the most appropriate for the data, Tucker or CP, see Figure 3.4.

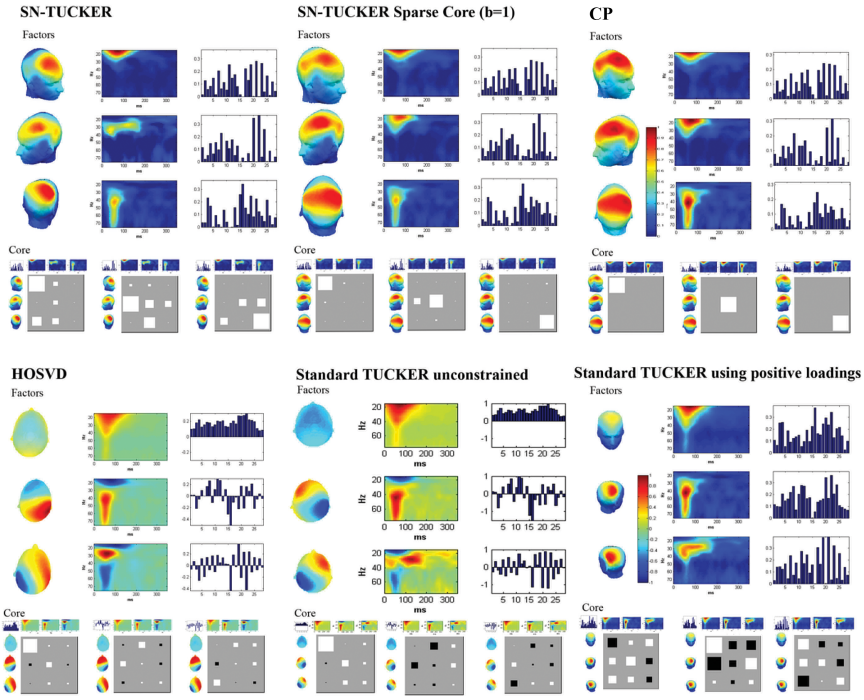


Figure 3.4: Analysis of the Inter Trial Phase Coherence (ITPC) data of EEG consisting of 14 subjects undergoing weight change of left hand during odd trials and right hand during even trials (for details on this data and the ITPC measure consult [Mørup et al., 2006b]). **Top left panel:** Example of result obtained when analyzing the data using SN-TUCKER. **Top middle panel:** Result when imposing sparseness on the core (sparsity strength $\beta = 1$, range of data [0;0.4]). **Top right panel:** The results obtained from the non-negative CP model corresponding to a fixed core having ones along the diagonal. **Bottom left panel:** The results obtained using HOSVD. **Bottom middle panel:** Results obtained using the Standard Tucker procedure provided by the N-way toolbox [Bro and Andersson, 2000] without constraints. **Bottom right panel:** Results obtained when imposing non-negativity on all the loadings but not on the core. Notice how the fully non-negative decomposition gives a more part based representation since no cancellation between components are allowed. Notice also how imposing sparseness on the core makes the Tucker decomposition resemble the CP decomposition.

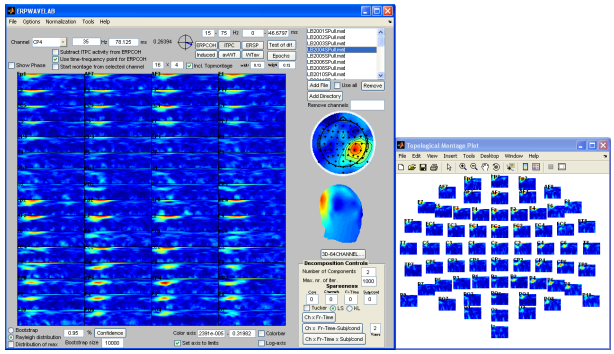


Figure 3.5: Screen shot of `www.erpwavelab.org` a toolbox for multi-channel analysis of wavelet transformed event related EEG data. One of the core features of the program is decomposition and visualization of the data through Non-negative Matrix and Tensor Factorization (NMF and NTF). For a thorough description of the toolbox see [Mørup et al., 2007a].

3.5 Other Tensor Decomposition models

In a sense the CP and Tucker model can be considered two extremes. The CP restricting the core to be diagonal, the Tucker the core to be arbitrary. By imposing constraints on the core a whole range of model between the two can be derived. However, a multitude of tensor decompositions that can not be considered simple restrictions on the Tucker core also exist. We will presently consider but a few of these.

3.5.1 PARAFAC2

The PARAFAC2 model was proposed in [Harshman, 1972b] and reads

$$\boldsymbol{X}^{(k)} \approx \boldsymbol{U}^{(k)} \operatorname{diag}(\boldsymbol{S}_{k,:}) \boldsymbol{V}^\top$$

where $\boldsymbol{U}^{(k)} \in \mathbb{R}^{I_1^{(k)} \times D}$. Thus, the collection of data $\boldsymbol{X}^{(k)}$ can vary in size across the first mode. PARAFAC2 is not unique without additional constraints since

$$\boldsymbol{U}^{(k)} \operatorname{diag}(\boldsymbol{S}_{k,:}) \boldsymbol{V}^\top = (\boldsymbol{U}^{(k)} \operatorname{diag}(\boldsymbol{S}_{k,:}) \boldsymbol{Q}^{-1} \boldsymbol{D}^{-1}) \boldsymbol{D} (\boldsymbol{V}\boldsymbol{Q})^\top = \widehat{\boldsymbol{U}}^{(k)} \operatorname{diag}(\widehat{\boldsymbol{S}}_{k,:}) \widehat{\boldsymbol{V}}^\top$$

Consequently, to improve the uniqueness properties, Harshman [1972b] imposed the constraint $\boldsymbol{U}^{(1)^\top}\boldsymbol{U}^{(1)} = \boldsymbol{U}^{(2)^\top}\boldsymbol{U}^{(2)} = \ldots = \boldsymbol{U}^{(K)^\top}\boldsymbol{U}^{(K)}$ requiring the components of the first mode to have same covariance. The benefit of the

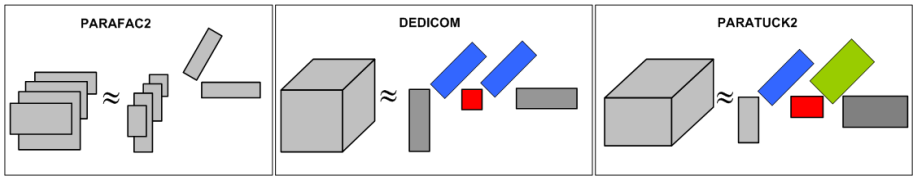


Figure 3.6: **Left panel:** The PARAFAC2 model allow data to vary in size over the first modality while decomposing the data into loadings such that each index of the third modality have specific loadings pertaining to the first modality such that only the covariance of these loadings have to be consistent. **Middle panel:** The DEDICOM model models an asymmetric relationship between the first and second modality of the data. Thus, the loadings of first and second mode are identical while the matrix indicated in red captures their asymmetric relationship. **Right panel:** The PARATUCK2 model also models an asymmetric relationship here between two potentially different types of modalities while the number of components used for the first and second modality can differ. Notice, two types of loadings are here used to model the third modality. Considering the blue, red and green matrices as one large core array the model corresponds to the Tucker2 model.

PARAFAC2 model is that the dimension of one mode can vary over another mode, hence, the model relaxes the tensorial structure of the data. Furthermore, components across the first mode can vary in structure for instance due to component shifts or warping as long as the covariance of the components remain the same. Thus, the PARAFAC2 model adds important flexibility to the CP model [Bro et al., 1999] while uniqueness of the decomposition is still guaranteed.

3.5.2 DEDICOM, Tucker2 and PARATUCK2

Decomposition into directional components (DEDICOM) was proposed in [Harshman, 1978]. The model is given by

$$\boldsymbol{X}^{(k)} = \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{D}_{k,:})\boldsymbol{R}\,\mathrm{diag}(\boldsymbol{D}_{k,:})\boldsymbol{A}^{\top}.$$

The model is motivated by data that reflects an asymmetric relationship between two identical modalities. For instance $\boldsymbol{X}^{(k)}$ could be a document-document link matrix and k index over time. Then $\boldsymbol{A}$ would denote document groups that are asymmetrically related through $\boldsymbol{R}$ such that each component $\boldsymbol{A}$ is present to degree $\boldsymbol{D}$ across time. The model has been generalized to the PARATUCK2 model [Harshman and Lundy, 1996] given by

$$\boldsymbol{X}^{(k)} = \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{S}_{k,:})\boldsymbol{R}\,\mathrm{diag}(\boldsymbol{T}_{k,:})\boldsymbol{B}^{\top}$$

Hence the model considers instead interactions between two possible different sets of objects where $\boldsymbol{A} \in \mathbb{R}^{I_1 \times D}$ and $\boldsymbol{B} \in \mathbb{R}^{I_2 \times R}$ pertain to each set of objects respectively and $\boldsymbol{R} \in \mathbb{R}^{D \times R}$ denotes their asymmetric relationship. The name is derived from the fact that this decomposition can be considered as a combination of CP and the so-called Tucker2 model, where the Tucker2 model is given by the Tucker model having one of the loadings set to identity.

Richard A. Harshman (1943-2008)

Richard Harshman was a great innovator of tensor decomposition and has fathered several important models apart from the PARAFAC model also the PARAFAC2 model, DEDICOM model, PARATUCK2 model and the ShiftCP model (see next chapter). Furthermore, he was one of the first to propose Latent Semantic Indexing [Deerwester et al., 1990].

3.6 Summary

While a multitude of tensor decomposition models have been presented, we have also seen that for these types of decompositions - simplicity of the representation is an important guiding principle. Furthermore, we have demonstrated how techniques from matrix analysis through the matrizicing, Khatri-Rao and Kronecker product operations can be used for the analysis of data with tensorial structure. Tensor decomposition is a field of growing attention and there are still many open problems to solve ranging from finding the best rank approximation, generalizing properties of the SVD to tensors and resolve the issues in terms of degeneracy in the CP model as well as correctly incorporating the tensorial structure of the data when imposing constraints such as independence.

Chapter 4

Decomposition with Invariance

> Plato's Theory of Forms asserts that Forms (or Ideas), and not the material world of change known to us through sensation, possess the highest and most fundamental kind of reality.
> *(source: www.Wikipedia.org)*
> *"It's all quite loopy from a modern perspective. But if your strip away the high-flown metaphysics, you can see that he* (Plato) *was really talking about invariance. His system of explanation was wildly off the mark, but his intuition that this was one of the most important questions we can ask about our own nature was a bull's eye"*
> - *Jeff Hawkins* "On Intelligence"

In this chapter we will consider extensions of the factor analysis model incorporating various types of invariance. Often the factor model is too restrictive and extensions are required to well capture the underlying structure of the data. One such common violation of the factor model is shifts in the component profiles due, for instance, to time delays between the sources and receivers. This is the focus of section 4.1. Another violation is reverberation such that the sources are repeated in some systematic way. This is addressed in the convolutive models of section 4.2. Finally, the data can be formed through sources that are manipulated through more general transformations than shifts. This will be treated in section 4.3 where we will pay particular attention to invariance of 2D rotation.

An important algorithmic framework for this chapter is the Time-Frequency Gradient Method TFGM described in Box 4 we proposed in [Mørup et al., 2007c] and formalized in [Madsen et al., 2008]. The method forms a framework to perform optimization in the time domain while taking advantage of model representation in the frequency domain.

Box 4: Time-Frequency Gradient Method (TFGM)
For objective functions, C, of the form

$$C = \sum_t f_t(x_t) + \frac{1}{J}\sum_f g_f(\widetilde{x}_f), \tag{4.1}$$

where f_t and g_f are real valued functions of the real and complex variables x_t and $\widetilde{x}_f$ such that $\widetilde{\mathbf{x}} = \mathcal{F}(\mathbf{x})$ the gradient with respect to x_t and $\widetilde{x}_f$ can be written as

$$\begin{aligned}\frac{\partial C}{\partial x_t} &= f_t'(x_t) + \frac{1}{J}\sum_f g_f'(\widetilde{x}_f)e^{i2\pi\frac{f}{J}t} = f_t'(x_t) + \mathcal{F}^{-1}(\mathbf{g}')_t,\\ \frac{\partial C}{\partial \widetilde{x}_f} &= \sum_t f'(x_t)e^{-i2\pi\frac{f}{J}t} + g_f'(\widetilde{x}_f) = \mathcal{F}(\mathbf{f}')_f + g_f'(\widetilde{x}_f).\end{aligned}$$

Thus, the gradients of C can be converted arbitrarily between the time and frequency domain.

Due to Parseval's identity the following holds

$$\sum_{i,j}\|\mathbf{x}_j - \boldsymbol{A}\mathbf{s}_j\|_F^2 = \tfrac{1}{J}\sum_{i,f}\|\widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}\widetilde{\mathbf{s}}_f\|_F^2. \tag{4.2}$$

Thus, the least square error in the time and frequency domain both have the form given in (4.1). For convenience, we presently index the zero frequency $\widetilde{\mathbf{x}}_0$.

4.1 Shifted Models

Assuming a reverberation-free environment with propagation delays form the Shifted Factor Analysis (SFA) model

$$\boldsymbol{X}_{i,j} \approx \sum_d \boldsymbol{A}_{i,d}\boldsymbol{S}_{d,j-\boldsymbol{\tau}_{i,d}}. \tag{4.3}$$

In [Harshman et al., 2003a] strong support was found for the conjecture that the incorporation of shifts can strengthen the model enough to make the parameters identifiable up to scaling and permutation (essential uniqueness). As demonstrated in Figure 4.3 this conjecture is not correct when allowing for arbitrary shifts. Indeed, the model is, as for regular factor analysis, ambiguous. In the complex domain the model reads

$$\widetilde{\mathbf{x}}_f \approx \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\mathbf{s}}_f. \tag{4.4}$$

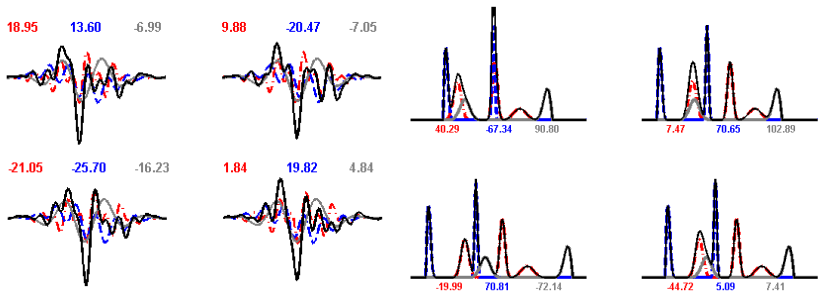
where $\widetilde{\boldsymbol{A}}^{(f)} = \boldsymbol{A} \bullet e^{-i2\pi \frac{f}{J} \boldsymbol{\tau}}$. Hence, the alternative model $\widetilde{\boldsymbol{A'}}^{(f)} = \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\boldsymbol{W}}^{(f)}$, $\widetilde{\mathbf{s}}'_f = \widetilde{\boldsymbol{W}}^{(f)^{-1}} \widetilde{\mathbf{s}}_f$ gives just as good an approximation, i.e. $\widetilde{\mathbf{x}}_f \approx \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\mathbf{s}}_f = \widetilde{\boldsymbol{A'}}^{(f)} \widetilde{\mathbf{s}}'_f$. Such that $\widetilde{\boldsymbol{W}}^{(f)} = \boldsymbol{W} \bullet e^{-i2\pi \frac{f}{J} \hat{\boldsymbol{\tau}}}$ is an invertible rotation, scaling and shift matrix.

Assume the inverse of $\widetilde{\boldsymbol{W}}^{(f)}$ is also a rotation, scaling and shift matrix, i.e. $\widetilde{\boldsymbol{W}}^{(f)^{-1}} = \boldsymbol{V} \bullet e^{-i2\pi \frac{f}{J} \check{\boldsymbol{\tau}}}$. Since $\widetilde{\boldsymbol{W}}^{(f)} \widetilde{\boldsymbol{W}}^{(f)^{-1}} = \boldsymbol{I}$, we find

$$\sum_{d''} \boldsymbol{W}_{d,d''} \boldsymbol{V}_{d',d''} e^{-i2\pi \frac{f}{J} (\hat{\boldsymbol{\tau}}_{d,d''} + \check{\boldsymbol{\tau}}_{d',d''})} = \begin{cases} 0 \text{ for } d \neq d' \forall\ f \\ 1 \text{ for } d = d' \forall\ f \end{cases}$$

From $f = 0$ we obtain the relation $\boldsymbol{V} = \boldsymbol{W}^{-1}$. For the remaining frequencies this expression can only be valid if $\hat{\boldsymbol{\tau}}_{dd''} + \check{\boldsymbol{\tau}}_{d''d} = 0$ (diagonal elements) and $\hat{\boldsymbol{\tau}}_{dd''} + \check{\boldsymbol{\tau}}_{d''d'} = k_{dd'}$ (off diagonal elements) where $k_{dd'}$ denotes an arbitrary constant. The first relation gives the constraint that $\hat{\boldsymbol{\tau}} = -\check{\boldsymbol{\tau}}^T$. The second relation further constraints all the elements of the columns of $\hat{\boldsymbol{\tau}}$ to be equal. Thus the ambiguity is given by $\widetilde{\boldsymbol{W}}^{(f)} = [\boldsymbol{W}\, \mathrm{diag}(e^{-i2\pi \frac{f}{J} \widehat{\boldsymbol{\tau}}})]$. Where $\widehat{\boldsymbol{\tau}}$ is a vector describing the shift ambiguity while $\boldsymbol{W}$ gives the ambiguity in terms of mixing strengths.

Prior to the work of [Harshman et al., 2003a,b] Bell and Sejnowski [Bell and Sejnowski, 1995] sketched how to handle time delays in networks based on a model similar to equation 4.3. This was further explored in [Torkkola, 1996]. In [Emile and Comon, 1998] a different model based on equally mixed sources, i.e. $\boldsymbol{A} = \mathbf{1}$, formed by moving averages incorporated non-integer delays by signal interpolation. Yeredor [Yeredor, 2003] solved the SFA model by joint diagonalization of the source cross spectra based on the AC-DC algorithm with non-integer shifts for the 2×2 system. This approach was extended to complex signals in [Yeredor, 2005].



(a) shifted ICA problem (b) shifted NMF problem

Figure 4.1: **Left panel:** Example of activities obtained (black graph) when summing three components (gray, blue dashed and red dash-dotted graphs) each shifted to various degrees (given in samples by the colored numbers). Clearly, the resulting activities are heavily impacted by the shifts such that a regular instantaneous factor analysis based decomposition would be inadequate. **Right panel:** Example of activities obtained (black graph) when summing three non-negative components (gray, blue dashed and red dash-dotted graphs) each shifted to various degrees (given in samples by the colored numbers). Clearly, the resulting activities are again heavily impacted by the shifts such that a regular instantaneous analysis of these non-negative components also would be inadequate.

4.1.1 Shift Invariant Subspace Analysis

In the following we describe the Shift Invariant Subspace Analysis (SISA) proposed in [Mørup et al., 2007b]. SISA finds a subspace that accounts for as much of the variation possible in the SFA model similarly to the SVD of regular factor analysis. The SISA is based on alternatingly solving for $\boldsymbol{A}$, $\boldsymbol{S}$ and $\boldsymbol{\tau}$.

A update:
Let $\widetilde{\boldsymbol{S}}_{d,f}^{(n)}$ denote the delayed version of the source signal $\widetilde{\boldsymbol{S}}_{d,f}$ to the i^{th} channel, i.e. $\widetilde{\boldsymbol{S}}_{d,f}^{(n)} = \widetilde{\boldsymbol{S}}_{d,f} e^{-i2\pi \frac{f}{J} \boldsymbol{\tau}_{i,d}}$. Then the SFA model can be restated as

$$\boldsymbol{X}_{i,:} \approx \boldsymbol{A}_{i,:} \boldsymbol{S}^{(i)}, \tag{4.5}$$

This is the regular factor analysis problem giving the update

$$\boldsymbol{A}_{i,:} = \boldsymbol{X}_{i,:} \boldsymbol{S}^{(i)\dagger}. \tag{4.6}$$

S update:
According to equation 4.4, $\mathbf{s}_f$ can be estimated as

$$\widetilde{\mathbf{s}}_f = \widetilde{\boldsymbol{A}}^{(f)^\dagger} \widetilde{\mathbf{x}}_f. \tag{4.7}$$

Although, $\boldsymbol{S}$ is updated in the frequency domain the updated version has to remain real when taking the inverse DFT. For $\boldsymbol{S}$ to be real valued the following has to hold

$$\widetilde{\mathbf{s}}_{J-f} = \widetilde{\mathbf{s}}_f^*, \tag{4.8}$$

where * denotes complex conjugate. This constraint is enforced by updating the first $\lfloor J/2 \rfloor + 1$ elements, i.e. up to the Nyquist frequency, while setting the remaining elements according to equation 4.8.

Non-integer delays in $\boldsymbol{\tau}$ can be estimated through the Newton-Raphson procedure described in Box 4.1.1a. However, this iterative update is sensitive to local minima, see Figure 4.2. Thus, to improve the algorithm from being stuck in suboptimal solutions integer shifts for $\boldsymbol{\tau}$ is estimated by the cross-correlation procedure given in Box 4.1.1b.

Box 4.1.1a: Newton-Raphson update for $\boldsymbol{\tau}$
The least squares error for the model stated in equation 4.4, is given by

$$C_{ls} = \frac{1}{M}\sum_f (\widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}^{(f)}\widetilde{\mathbf{s}}_f)^H(\widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}^{(f)}\widetilde{\mathbf{s}}_f), \tag{4.9}$$

where ^{H} denotes the conjugate transpose. Define $\boldsymbol{T}^{ID\times 1} = vec(\boldsymbol{\tau})$, i.e. the vectorized version of the matrix $\boldsymbol{\tau}$ such that $\boldsymbol{T}_{i+(d-1)I} = \boldsymbol{\tau}_{i,d}$. Let further

$$\widetilde{\mathcal{Q}}_{i,d,f} \quad = \quad \widetilde{\boldsymbol{A}}^{(f)}_{i,d}\widetilde{\boldsymbol{S}}_{d,f}, \quad \widetilde{\mathbf{e}}_f = \widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}^{(f)}\widetilde{\mathbf{s}}_f. \tag{4.10}$$

Then the gradient of C_{ls} with respect to $\boldsymbol{\tau}_{i,d}$ is given as

$$\boldsymbol{g}_{i+(d-1)I} = \frac{\partial C_{ls}}{\partial \boldsymbol{T}_{i+(d-1)I}} = \frac{\partial C_{ls}}{\partial \boldsymbol{\tau}_{i,d}} = \frac{-1}{J}\sum_f 2\omega\Im[\widetilde{\mathcal{Q}}_{i,d,f}\widetilde{\boldsymbol{E}}^*_{n,f}]$$

The Hessian has the following structure

$$\boldsymbol{H}_{i+(d-1)I,i'+(d'-1)I} = \begin{cases} \frac{-2}{J}\sum_f \omega^2\Re[\widetilde{\mathcal{Q}}_{i,d,f}\widetilde{\mathcal{Q}}^*_{i',d',f}] \text{ if } \mathrm{i} \neq \mathrm{i}' \wedge \mathrm{d} \neq \mathrm{d}' \\ \frac{-2}{J}\sum_f \omega^2\Re[\widetilde{\mathcal{Q}}_{i,d,f}(\widetilde{\mathcal{Q}}^*_{i',d',f} + \widetilde{\boldsymbol{E}}^*_{i',f})] \text{ if } \mathrm{i} = \mathrm{i}' \wedge \mathrm{d} = \mathrm{d}' \end{cases}$$

As a result, $\boldsymbol{\tau}$ can be estimated using the Newton-Raphson method

$$\boldsymbol{T} \leftarrow \boldsymbol{T} - \eta\boldsymbol{H}^{-1}\boldsymbol{g},$$

where η is a step size parameter that is tuned to keep decreasing the cost function.

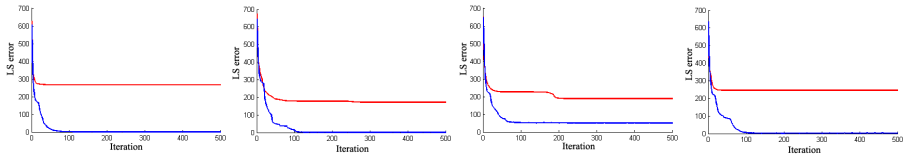


Figure 4.2: Four runs illustrating how using the cross-correlation (CC) procedure every 20^{th} iteration in combination with the Newton-Raphson (NR) update (blue curve) rather than relying solely on the NR update to estimate the delays (red curve) improves the algorithm from getting stuck in suboptimal solutions. Whereas the NR method alone does not identify the global minima, this is found in 3 out of the 4 trials when re-estimating the delays using the CC procedure. Notice, how the sudden jumps in the curves using the CC procedure are initiated at the iterations where the cross-correlation was used to re-estimate the delays.

Box 4.1.1b: Cross-correlation procedure for $\boldsymbol{\tau}$ update

Let

$$\widetilde{\boldsymbol{R}}_{i,f} = \widetilde{\boldsymbol{X}}_{i,f} - \sum_{d \neq d'} \widetilde{\boldsymbol{A}}^{(f)}_{i,d} \widetilde{\boldsymbol{S}}_{d,f},$$

i.e., the signal at the i^{th} sensor at frequency f when projecting all but the d' source out of $\widetilde{\boldsymbol{X}}$. Then the least squares objective can be written as

$$\frac{1}{2}\sum_{f} \|\widetilde{\mathbf{X}}_f - \widetilde{\boldsymbol{A}}^{(f)}\widetilde{\mathbf{s}}_{\mathbf{f}}\|_F^2 = \frac{1}{2}\sum_{i,f} \|\widetilde{\boldsymbol{R}}_{i,f} - \widetilde{\boldsymbol{A}}^{(f)}_{i,d'}\widetilde{\boldsymbol{S}}_{d',f}\|_F^2$$

$$= \frac{1}{2}\sum_{i,f} \|\widetilde{\boldsymbol{R}}_{i,f}\|_F^2 - 2\widetilde{\boldsymbol{R}}^*_{i,f}\boldsymbol{A}_{i,d'}\widetilde{\boldsymbol{S}}_{d',f}e^{-i2\pi\frac{f}{J}\tau_{i,d'}} + \|\boldsymbol{A}_{i,d'}\widetilde{\boldsymbol{S}}_{d',f}\|_F^2$$

Thus, the optimal value of $\boldsymbol{\tau}_{i,d'}$ is given as the maximal absolute cross-correlation between the d' source and i^{th} sensor given as $\widetilde{\boldsymbol{c}}_f = \widetilde{\boldsymbol{R}}^*_{i,f}\widetilde{\boldsymbol{S}}_{d',f}$. Hence, $\boldsymbol{\tau}_{i,d'}$ can be estimated as

$$\begin{aligned} \boldsymbol{\tau}_{i,d'} &= t - (J+1) \quad \text{where} \quad t = \arg\max_j |\boldsymbol{c}_j|, \\ \boldsymbol{A}_{i,d'} &= \frac{\boldsymbol{c}_t}{\boldsymbol{S}_{d',:}\boldsymbol{S}^T_{d',:}}. \end{aligned}$$

The corresponding value of $\boldsymbol{A}_{i,d'}$ is also given above.

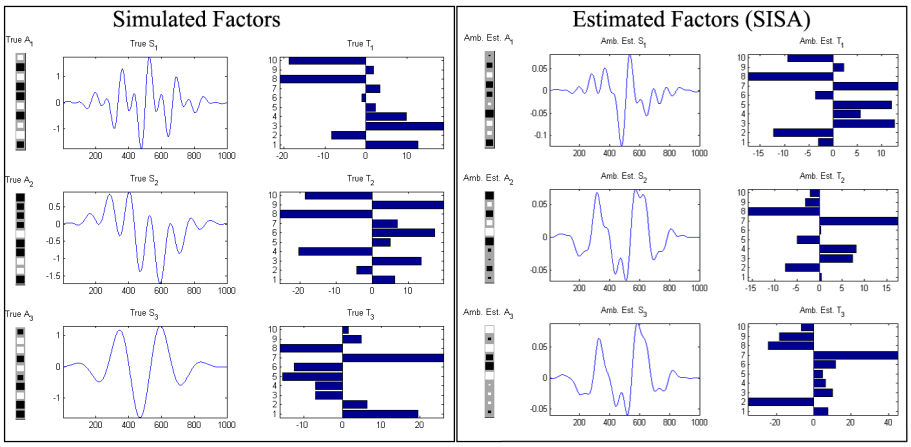


Figure 4.3: Results obtained by a shift invariant subspace analysis (SISA). **Left panel:** The true factors forming a synthetic data set. To the left, the strength of the mixing $\boldsymbol{A}$ of each source is indicated such that the mixing strength is given by the size of the white and black areas where white is positive and black negative mixing. In the middle, the three sources are shown and to the right is given the time delays of each source to each channel. **Right panel:** The estimated factors obtained by SISA. Although, all the variance is explained the decomposition has not identified the true underlying components but an ambiguous mix. Clearly, as for regular factor analysis the Shifted Factor Analysis model is not unique thus additional constraints are required to resolve the ambiguities inherent in the model.

4.1.2 Shifted Independent Component Analysis

As for regular factor analysis the ambiguities of the Shifted Factor Analysis (SFA) can be resolved imposing independence on the sources forming the shifted independent component analysis (SICA). We will here consider the maximum likelihood (ML) method for imposing independence. In the framework of ML a non-gaussian distribution on the sources is assumed such that ambiguity can be resolved up to the trivial ambiguities of scale, permutation and source shifting relative to the time delays, see also section 2.3.

Define, $\widetilde{\mathbf{u}}_f = \widetilde{\boldsymbol{W}}^{(f)}\widetilde{\mathbf{s}}_f$, i.e. the sources at frequency f when transformed according to the rotation and shift ambiguity described in section 4.1. The ambiguity can be resolved by maximizing the log-likelihood assuming the (non-gaussian)

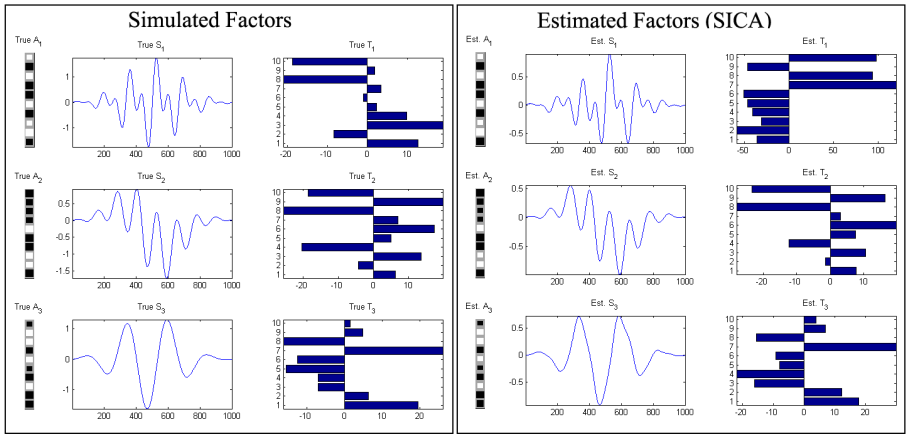


Figure 4.4: Result obtained using the SICA on the decomposition found using SISA. By imposing independence, e.g., requiring the amplitudes in the frequency domain to be sparse, the rotation and shift ambiguity inherited in the model is resolved. Clearly, the true underlying components and their respective mixing are correctly identified. However, a local minima has been found, resulting in errors in the estimation of the delays particularly for the first component.

Laplace distribution $p(\widetilde{\mathbf{u}}_{d,f}) \propto e^{-|\widetilde{U}_{d,f}|}$, i.e.

$$p(\widetilde{\mathbf{s}}_f|\boldsymbol{W},\widehat{\boldsymbol{\tau}}) = \prod_f p(\widetilde{\mathbf{s}}_f|\boldsymbol{W},\widehat{\boldsymbol{\tau}}) = \prod_f |det(\widetilde{\boldsymbol{W}}^{(f)})|p(\widetilde{\boldsymbol{W}}^{(f)}\widetilde{\mathbf{s}}_f)$$

Such that the log-likelihood as a function of $\boldsymbol{W}$ and $\widehat{\boldsymbol{\tau}}$ becomes

$$\mathcal{L}(\boldsymbol{W},\widehat{\boldsymbol{\tau}}) = \sum_f \ln|\det(\widetilde{\boldsymbol{W}}^{(f)})| - \sum_d |\widetilde{\boldsymbol{W}}^{(f)}\widetilde{\mathbf{s}}_f|_d$$

By maximizing $\mathcal{L}(\boldsymbol{W},\widehat{\boldsymbol{\tau}})$, $\boldsymbol{W}$ and $\widehat{\boldsymbol{\tau}}$ is estimated and a new unambiguous $\boldsymbol{S}$ solution found by $\widetilde{\mathbf{s}}_f^{SICA} = \widetilde{\boldsymbol{W}}^{(f)}\widetilde{\mathbf{s}}_f^{SISA}$. The corresponding mixing and delays can be estimated alternating between the $\boldsymbol{A}$ and $\boldsymbol{\tau}$ update given in section 4.1.1. We initialized $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{A}\boldsymbol{W}^{-1}$ and $\boldsymbol{\tau}_{i,d}$ by the cross-correlation procedure. In Figure 4.4 is given the components found resolving the ambiguities using the above SICA.

4.1.3 Shifted Non-negative Matrix Factorization

The Shifted Non-negative Matrix Factorization (ShiftNMF) proposed in [Mørup et al., 2007c] is given by

$$\boldsymbol{X}_{i,j} \approx \sum_d \boldsymbol{A}_{i,d} \boldsymbol{S}_{d,j-\boldsymbol{\tau}_{i,d}}.$$

where $\boldsymbol{X} \geq 0$, $\boldsymbol{A} \geq 0$ and $\boldsymbol{S} \geq 0$. The model was motivated in [Mørup et al., 2007c] considering the use of NMF on data such as magnetic resonance spectra [Du et al., 2005], data obtained using astronomical spectrometers for the identification and classification of space objects [Pauca et al., 2006] and data of fluorescence spectra [Gobinet et al., 2004]. For these types of data, the ShiftNMF model is potentially useful since shift in the spectral profiles often occurs for instance as a result of the Doppler effect. Furthermore, NMF has proven useful for extracting intensity profiles over time from biomedical data such as PET imaging [Lee et al., 2001; Kim et al., 2001; Bödvarsson et al., 2007]. Here, potential delays in the onset of the various profiles can also be handled by the ShiftNMF model.

The non-negativity constraint is not transparent in the frequency domain. Thus, a method combining the apparent representation of non-negativity in the time domain with the efficient implementation of shifts in the frequency domain is desired. We recall that the least squares objective of the ShiftNMF model can be written as

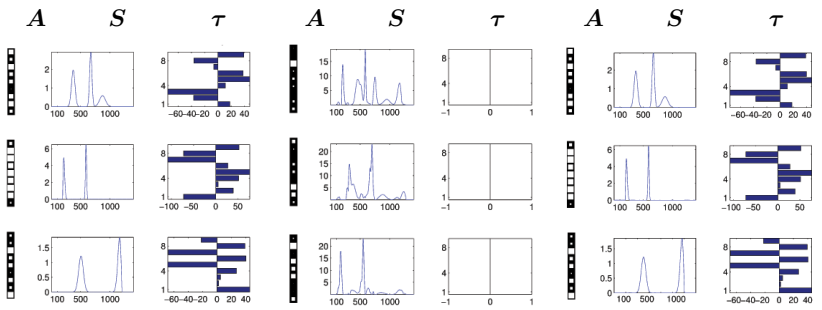
$$C_{LS}(\boldsymbol{A}, \boldsymbol{S}) = \frac{1}{2J} \sum_f \|\widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\mathbf{s}}_f\|_F^2.$$

Thus, in the frequency domain the objective is separable over frequencies such that we can use the TFGM given in Box 4 to combine the above efficient frequency representation with non-negativity constraints residing in the time domain. The gradient of the least squares cost function in the frequency domain is

$$\widetilde{\mathbf{g}}_f = \frac{\partial C_{LS}}{\partial \widetilde{\mathbf{S}}_f} = -\frac{1}{J} \widetilde{\boldsymbol{A}}^{(f)^H} (\widetilde{\mathbf{x}}_f - \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\mathbf{s}}_f).$$

By applying the IDFT on the gradient in the frequency domain the corresponding gradient in the time domain is obtained. Splitting the gradient in the frequency domain into what constitutes the positive and negative part of the corresponding gradient in the time-domain gives

$$\begin{aligned} \widetilde{\mathbf{g}}_f^+ &= \frac{1}{J} \widetilde{\boldsymbol{A}}^{(f)^H} \widetilde{\boldsymbol{A}}^{(f)} \widetilde{\mathbf{s}}_f, \\ \widetilde{\mathbf{g}}_f^- &= \frac{1}{J} \widetilde{\boldsymbol{A}}^{(f)^H} \widetilde{\mathbf{x}}_f. \end{aligned}$$



(a) True components (b) Est. comp. NMF (c) Est. comp. ShiftNMF

Figure 4.5: **Left panel:** The true factors forming the synthetic data ($\mathbf{X} \in \mathbb{R}^{9\times 1400}$). To the left, the strength of the mixing $\mathbf{A}$ of each source is given by the size of the white areas. In the middle, the three sources are shown and to the right is given the time delays of each source to each channel. **Middle panel:** Results obtained by conventional instantaneous NMF for the generated synthetic data. Clearly, the model cannot account for the shifts in the data hence the sources are incorrectly identified, thus, only 68 % of the variance of the data can be accounted for. **Right panel:** The estimated factors obtained by a ShiftNMF analysis. Clearly, the model with shifts has correctly recovered the components of the synthetic data hence accounts for all the variance in the data.

Consequently, by taking the IDFT of $\widetilde{\mathbf{G}}^+$ and $\widetilde{\mathbf{G}}^-$ the corresponding positive and negative part of the gradient in the time-domain are found. As a result, $\mathbf{S}$ can be updated using multiplicative updates in the time domain, hence, enforcing non-negativity through the update

$$\boldsymbol{S} = \boldsymbol{S} \bullet \left(\frac{\boldsymbol{G}^-}{\boldsymbol{G}^+} \right)^{.\alpha}.$$

$\boldsymbol{A}$ can be estimated using regular factor analysis according to (4.6) thus non-negativity can be imposed straight forward using for instance the NNQP given in Algorithm 2.4 or using multiplicative updates. $\boldsymbol{\tau}$ can be estimated using the Newton Raphson and Cross-correlation procedure given in Box 4.1.1a and 4.1.1b.

In Figure 4.5 we demonstrate the usefulness of the ShiftNMF over regular instantaneous NMF when shifts are present in the data.

4.1.4 Shifted Sparse Coding

The Shifted Sparse Coding model (ShiftSC) is given by

$$\boldsymbol{X}_{i,j} \approx \sum_d \boldsymbol{A}_{i,d} \boldsymbol{S}_{d,j-\boldsymbol{\tau}_{i,d}}.$$

where $\mathbf{S}$ is sparse. $\boldsymbol{A}$ and $\boldsymbol{\tau}$ can be updates as described in the section on SISA such that the components of $\boldsymbol{A}$ have unit l_2-norm based on the normalization invariance approach given in section A.1.3.2. We presently consider sparseness as imposed using the Laplace prior, i.e. given by a l_1-norm penalty term

$$C_{(A,S)} = \frac{1}{2J} \sum_f \|\widetilde{\mathbf{x}}_f - \widetilde{\mathbf{A}}^{(f)} \widetilde{\mathbf{s}}_f\|_F^2 + \lambda \|\boldsymbol{S}\|_1.$$

Thus, in the frequency domain the least squares error is again separable over frequencies, however, the sparseness constraint resides in the time domain. The gradient of the least squares error in the frequency domain is

$$\widetilde{\mathbf{g}}_f^{LS} = -\frac{1}{J} \widetilde{\mathbf{A}}^{(f)^H} (\widetilde{\mathbf{x}}_f - \widetilde{\mathbf{A}}^{(f)} \widetilde{\mathbf{s}}_f).$$

Using the TFGM we now find the gradient of the cost function including the l_1-norm penalty

$$\nabla_{\boldsymbol{S}} C = \mathcal{F}(\widetilde{\mathbf{G}}^{LS}) + \lambda \operatorname{sign}(\boldsymbol{S}).$$

As a result, $\boldsymbol{S}$ can be updated according to the GB-SC given in Algorithm 2.5.

4.1.5 Shifted CP

The Shifted Factor Analysis was generalized to the CP model in [Hong and Harshman, 2003; Mørup et al., 2008a]. The model was in [Mørup et al., 2008a] motivated considering NeuroImaging data. For NeuroImaging data it is customary, prior to or post averaging over trials, to decompose the data using various types of factor analysis based decompositions, i.e.

$$\boldsymbol{X}_{i,j} \approx \sum_d^D \boldsymbol{A}_{i,d} \boldsymbol{B}_{j,d}$$

where we for convenience here write $\boldsymbol{B}$ instead of $\boldsymbol{S}$ [Donchin and Heffley, 1978; Makeig et al., 1996, 1997, 2002; McKeown et al., 1998, 2003]. For Electroencephalography (EEG) and functional Magnetic Resonance Imaging (fMRI) data the recorded data can be represented by the channel/voxel by time matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J}$. The decomposition above then describes the data as a sum

of components separated into time profiles $\boldsymbol{B}_d$ with corresponding spatial topographies $\boldsymbol{A}_d$. However, since modeling the data by this factor analysis type decomposition is ambiguous additional constraints which are not necessarily physiologically justified have to be imposed. Forming a decomposition based on factor analysis where each component varies in strength over trials yields the CP model

$$\mathcal{X}_{i,j,k} \approx \sum_{d}^{D} \boldsymbol{A}_{i,d}\boldsymbol{B}_{j,d}\boldsymbol{C}_{k,d},$$

where $\boldsymbol{C}_d$ gives the degree in which the profile time series $\boldsymbol{B}_d$ with spatial topography $\boldsymbol{A}_d$ is present throughout the various trials. As described in chapter 3 the CP model is unique under mild conditions. Conditions that, in the presence of noise in the data, are practically always satisfied. Consequently, modeling repeated trials by CP in theory not only improves the component identification but also resolves the ambiguities encountered when modeling the data by (2-way). Notice, the application of CP to EEG was already suggested in the original paper on CP [Harshman, 1970] and later reinvented by Möcks [1988] naming the model topographic component analysis. In Andersen and Rayens [2004] it was further demonstrated how the CP model is useful in the analysis of neuroimaging data such as fMRI [Andersen and Rayens, 2004]. Additional applications of multi-way modeling in EEG and fMRI include [Möcks, 1988; Field and Graupe, 1991; Wang et al., 2000; Beckmann and Smith, 2005; Miwakeichi et al., 2004; Mørup et al., 2006a; De Vos et al., 2007; Acar et al., 2007].

Unfortunately, in the above analysis degenerate solutions as described in chapter 3 are known to occur. This makes the CP estimation unstable, slow in convergence and difficult to interpret since the components are dominated by strong cancelation effects between the various components in the model [Harshman and Lundy, 1984]. It has been argued in a number of papers that the degenerate solutions might be explained by components being delayed with respect to each other [Field and Graupe, 1991; Andersen and Rayens, 2004; Harshman et al., 2003a; Hong and Harshman, 2003]. To overcome degeneracy, orthogonality [Field and Graupe, 1991] or independence [Beckmann and Smith, 2005] has been imposed or the signal analyzed considering purely additive models based on analysis of amplitudes in a spectral representation [Miwakeichi et al., 2004; Mørup et al., 2006a]. However, if shifts are causing degeneracy it is more natural to extend the CP model to account for shifts rather than resorting to constraints such as orthogonality or independence that are not necessarily physiologically justified. Furthermore, the degeneracy into profiles which resemble pairs of functions and their derivatives such as pairs of cosine and sine functions as found in [Field and Graupe, 1991] gives strong indication that the data should be decomposed by a model accounting for shifts rather than models based on instantaneous mixing. Finally, for fMRI data shifts naturally occur due to hemodynamic delay [Buxton et al., 1998] while for EEG data onset changes of physiological activity

unrelated to the event such as eye blinks cause delays across the trials. The ShiftCP model (shifted over third mode) reads

$$\mathcal{X}_{i,j,k} \approx \sum_{d}^{D} \boldsymbol{A}_{i,d} \boldsymbol{B}_{j-\boldsymbol{\tau}_{k,d},d} \boldsymbol{C}_{k,d}$$

such that each time profile $\boldsymbol{B}_d$ is shifted according to the index of the third mode $\boldsymbol{\tau}_{k,d}$ time-samples. The ShiftCP model is illustrated in Figure 4.6.

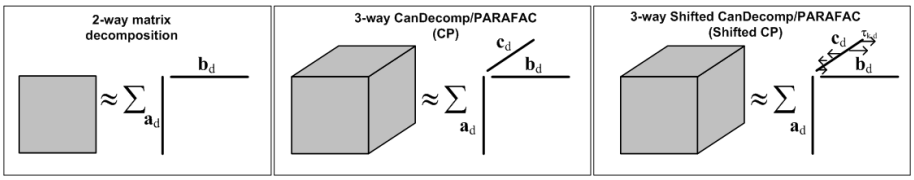


Figure 4.6: As described in chapter 3 the CP model can be considered a straightforward generalization of 2-way (matrix) decomposition (left panel) to arrays of more than two modalities (middle panel). Thus, the data is described by an outer product of factors pertaining to each of the modalities. The ShiftCP model extends the CP model to allow for shifts to occur over the second mode such that for each index of the third mode the component of the second mode is shifted a given amount (right panel).

In the frequency domain the ShiftCP model is given by

$$\widetilde{\mathcal{X}}_{i,f,k} \approx \sum_{d}^{D} \boldsymbol{A}_{i,d} \widetilde{\boldsymbol{B}}_{f,d} \boldsymbol{C}_{k,d} e^{-i2\pi \frac{f}{J} \boldsymbol{\tau}_{k,d}}.$$

Let $\widetilde{\boldsymbol{B}}^{(k)}_{f,d} = \widetilde{\boldsymbol{B}}_{f,d} \bullet e^{-i2\pi \frac{f}{J} \boldsymbol{\tau}_{k,d}}$, i.e. $\widetilde{\boldsymbol{B}}$ componentwise shifted according to the delays to the k^{th} channel. Let further $\mathbf{Z}_{j+k(J-1),d} = \boldsymbol{C}_{k,d} \boldsymbol{B}^{(k)}_{j,d}$, i.e. the Khatri-Rao product between $\boldsymbol{C}$ and the shifted version of $\boldsymbol{B}$.

Using the n-mode matricizing and the Khatri-Rao product we can as for regular CP state the estimation of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ as a sequence of ordinary factor analysis problems

$$\begin{aligned}
\boldsymbol{X}_{(1)} &\approx \boldsymbol{A}\boldsymbol{Z}^T & \Rightarrow \quad & \boldsymbol{A} \leftarrow \boldsymbol{X}_{(1)} \boldsymbol{Z}^{T^\dagger} \\
\widetilde{\boldsymbol{X}}_{(2)_{f,:}} &\approx \widetilde{\boldsymbol{B}}_{f,:} (\widetilde{\boldsymbol{C}}^{(f)} \odot \boldsymbol{A})^T & \Rightarrow \quad & \widetilde{\boldsymbol{B}}_{f,:} \leftarrow \widetilde{\boldsymbol{X}}_{(2)_{f,:}} (\widetilde{\boldsymbol{C}}^{(f)} \odot \boldsymbol{A})^{T^\dagger} \\
\boldsymbol{X}_{(3)_{k,:}} &\approx \boldsymbol{C}_{k,:} (\boldsymbol{B}^{(k)} \odot \boldsymbol{A})^T & \Rightarrow \quad & \boldsymbol{C}_{k,:} \leftarrow \boldsymbol{X}_{(3)_{k,:}} (\boldsymbol{B}^{(k)} \odot \boldsymbol{A})^{T^\dagger}
\end{aligned}$$

Since the estimation is stated as regular factor analysis problems non-negativity constraints for $\boldsymbol{A}$ and $\boldsymbol{C}$ can be imposed for instance using the NNQP given

in Algortihm 2.4. As the estimation is based on regular factor analysis it is also straight forward to derive the corresponding ShiftCP algorithms forming a shifted Non-negative Tensor Factorization (ShiftNTF) as well as a Shifted Sparse Coding model for tensors.

For the update of $\boldsymbol{\tau}$ the following cross-correlation similar to the procedure derived for the SISA given in Box 4.1.1b can be derived. Let

$$\boldsymbol{R}_{(3)_{k,:}^{d'}} = \boldsymbol{X}_{(3)_{k,:}} - \sum_{d \neq d'} \boldsymbol{C}_{k,d} (\boldsymbol{B}_d^{(k)} \odot \boldsymbol{A}_d)^T,$$

i.e. $\boldsymbol{R}_{(3)_{k,:}}^{d'}$ is the remaining signal at the k^{th} row when projecting all but the $d'th$ source out of $\boldsymbol{X}_{(3)}$. Notice, with this notation the least squares error can be rewritten as

$$\begin{aligned}\sum_k \|\boldsymbol{X}_{(3)_{k,:}} - \sum_d^D \boldsymbol{C}_{k,d} (\boldsymbol{B}_d^{(k)} \odot \boldsymbol{A}_d)^T)\|^2 &= \sum_k \|\boldsymbol{R}_{(3)_{k,:}}^{d'} - \boldsymbol{C}_{k,d'} (\boldsymbol{B}_{d'}^{(k)} \odot \boldsymbol{A}_{d'})^T\|^2 \\ = \|\boldsymbol{R}_{(3)_{k,:}}^{d'}\|^2 - \boldsymbol{C}_{k,d'} \sum_j \boldsymbol{B}_{j-\boldsymbol{\tau}_{k,d'},d'} \sum_i \mathcal{R}_{i,j,k}^{d'} \boldsymbol{A}_{i,d'} &+ \|\boldsymbol{C}_{k,d'} (\boldsymbol{B}_{d'}^{(k)} \odot \boldsymbol{A}_{d'})^T\|^2.\end{aligned}$$

The first and third term is independent of $\boldsymbol{\tau}_{k,d'}$. Thus, the least square error is minimized when the second term is maximized. Since $\boldsymbol{C}_{k,d'}$ is a constant this can be omitted such that we get

$$\begin{aligned}\mathbf{r}_j^{(k,d')} &= \sum_i \mathcal{R}_{i,j,k}^{d'} \boldsymbol{A}_{i,d'} \\ \widetilde{\mathbf{c}}_{k,d'}(f) &= \widetilde{\mathbf{r}}_f^{(k,d')^*} \widetilde{\boldsymbol{B}}_{f,d'}.\end{aligned}$$

$\boldsymbol{\tau}_{k,d'}$ can now be estimated as

$$\begin{aligned}\widehat{\boldsymbol{\tau}}_{k,d'} &= \arg\max_t |\mathbf{c}_{k,d'}(t)| \\ \boldsymbol{\tau}_{k,d'} &= \widehat{\boldsymbol{\tau}}_{k,d'} - (J+1).\end{aligned}$$

I.e. as the delay corresponding to maximum absolute cross-correlation between $\mathbf{r}^{(k,d')}$-the time profile of the residual for the d' component and $\mathbf{b}_{d'}$-the component time profile. The value of $\boldsymbol{C}_{k,d'}$ corresponding to this delay is given by

$$\boldsymbol{C}_{k,d'} = \frac{\boldsymbol{c}_{k,d'}(\widehat{\boldsymbol{\tau}}_{k,d'})}{\boldsymbol{B}_{d'}^T \boldsymbol{B}_{d'}}.$$

If $\boldsymbol{C}$ is constrained positive only positive values of $\boldsymbol{c}_{k,d}(t)$ is considered. The above procedure can only estimate integer delays. However, by minimizing the least squares error in the complex domain with respect to $\boldsymbol{\tau}$ a gradient and Hessian can be calculated such that non-integer delays can be estimated for instance by the Newton-Raphson procedure given in Box 4.1.1a.

The number of components for the ShiftCP model can for instance be estimating using the *CorConDiag* given in section 3.3 based on the following Tucker extension of the ShiftCP model

$$\mathcal{X}_{i_1,i_2,i_3} \approx \sum_{d_1,d_2,d_3} \mathcal{G}_{d_1,d_2,d_3} \boldsymbol{A}^{ShiftCP}_{i_1,d_1} \boldsymbol{B}^{ShiftCP}_{i_2-\boldsymbol{\tau}^{ShiftCP}_{i_3,d_2},d_2} \boldsymbol{C}^{ShiftCP}_{i_3,d_3}.$$

Notice, the ShiftCP model is no longer by itself trilinear and as such evaluating further deviation from trilinear structure in the Core may seem questionable, however, in practice this approach worked well on controlled experiments using synthetic data [Mørup et al., 2008a]. Unfortunately, the rigorous proof of uniqueness by Kruskal using k-rank given in equation 3.1 is involved. However, the uniqueness assuming $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ all of full rank can be proven by considering the CP model in a slab representation as described in chapter 3. For the k^{th} slab the CP model reads

$$\begin{aligned} \boldsymbol{X}_{(:,:,k)} \approx \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{B}^T &= (\boldsymbol{AT})(\boldsymbol{T}^{-1}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{Q})(\boldsymbol{Q}^{-1}\boldsymbol{B}^T) \\ &= \widehat{\boldsymbol{A}}\,\mathrm{diag}(\widehat{\boldsymbol{C}}_{k,:})\widehat{\boldsymbol{B}}^{\top}. \end{aligned}$$

Thus, if two solutions $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}, \widehat{\boldsymbol{C}}$ exists we recall that there must be a mapping from one solution to the other given by $\boldsymbol{T}$ and $\boldsymbol{Q}$. However, for this mapping the term $\boldsymbol{T}^{-1}diag(\boldsymbol{C}_{k,:})\boldsymbol{Q}$ has to be diagonal for all k which when $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ have full rank restricts $\boldsymbol{T}$ and $\boldsymbol{Q}$ to be simple scale and permutation matrices [Harshman, 1972a; Möcks, 1988]. For the ShiftCP model we instead have

$$\begin{aligned} \boldsymbol{X}_{(:,:,k)} \approx \boldsymbol{A}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{B}^{(k)^T} &= (\boldsymbol{AT})(\boldsymbol{T}^{-1}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{Q})(\boldsymbol{Q}^{-1}\boldsymbol{B}^{(k)^T}) \\ &= \widehat{\boldsymbol{A}}\,\mathrm{diag}(\widehat{\boldsymbol{C}}_{k,:})\widehat{\boldsymbol{B}}^{(k)^T} \end{aligned}$$

Where $\boldsymbol{B}^{(k)}_d = \boldsymbol{B}_{j-\boldsymbol{\tau}_{k,d},d}$. Although, the CP model is extended such that $\boldsymbol{B}$ is shifted $(\boldsymbol{T}^{-1}\,\mathrm{diag}(\boldsymbol{C}_{k,:})\boldsymbol{Q})$ still has to remain diagonal for all values of k. This again strongly restricts $\boldsymbol{T}$ and $\boldsymbol{Q}$. The obvious ambiguities are scaling, permutation, relative shift and onset as well as period of the time-series. In [Mørup et al., 2008a] we tested the uniqueness of the decomposition by investigating the similarity of 250 decompositions randomly initialized. We found that the 50 decompositions with lowest least square error were all identical.

4.2 Convolutive Models

While the shifted models of the previous section can account for specific delays between the sources and sensors the convolutive models of this section can take into account general reverberation of the sources.

4.2.1 Convolutive ICA

In for instance real acoustic environments we expect reverberations. To account for general delay mixing effects, the ICA model has been generalized to convolutive mixtures (cICA), see e.g., [Attias and Schreiner, 1998; Parra et al., 1998; Anemuller et al., 2003]

$$\boldsymbol{X}_{i,j} \approx \sum_{\tau,d} \boldsymbol{A}^{\tau}_{i,d} \boldsymbol{S}_{d,j-\tau}, \quad \boldsymbol{x}_j \approx \sum_{\tau} \boldsymbol{A}^{\tau} \boldsymbol{s}_{j-\tau}.$$

Here $\boldsymbol{A}^{\tau}$ is a filter that accounts for the presence of each source in the sensors at time delay τ. The Shifted Factor model, thus is a special case of the convolutive model where the filter coefficients $\boldsymbol{A}^{\tau}_{i,d} = \boldsymbol{A}_{i,d}$ if $\boldsymbol{\tau}_{i,d} = \tau$ else $\boldsymbol{A}^{\tau}_{i,d} = 0$. In the frequency domain this convolutive extension can be expressed as

$$\boldsymbol{X}(w) = \boldsymbol{A}(w)\boldsymbol{S}(w)$$

Thus, in the frequency domain the convolutive ICA problem becomes a regular ICA type decomposition for each frequency. However, due to scale and permutation ambiguity expressed through the matrix $\boldsymbol{P}(w)$ the solutions $\widehat{\boldsymbol{A}}(w) = \boldsymbol{A}(w)\boldsymbol{P}(w)^{-1}$ and $\widehat{\boldsymbol{S}}(w) = \boldsymbol{P}(w)\widehat{\boldsymbol{S}}(w)$ are equally good. Thus, there is an inherent problem relating the components obtained for each frequency to each other. A range of solutions have been proposed to solve for this scale and permutation problem based on optimizing for consistency of the filter coefficients or consistency of the spectrum of the recovered signals [Syskind et al., 2007]. Alternative methods have estimated the model in the time domain exploiting that the problem can be stated in terms of Toeplitz structured matrices. A thorough survey of convolutive ICA methods both in the time and frequency domain can be found in [Syskind et al., 2007].

4.2.2 Convolutive NMF

The convolutive NMF (cNMF) model first proposed in [Smaragdis, 2004] is given by

$$\boldsymbol{X}_{i,j} \approx \sum_{\tau,d} \boldsymbol{A}^{\tau}_{i,d} \boldsymbol{S}_{d,j-\tau}, \quad \mathbf{x}_j \approx \sum_{\tau} \boldsymbol{A}^{\tau} \mathbf{s}_{j-\tau}.$$

where $\boldsymbol{X} \geq 0$, $\boldsymbol{A} \geq 0$ and $\boldsymbol{S} \geq 0$. This model has been used to analyze the absolute spectrogram of music signals [Smaragdis, 2004] based on multiplicative updates. In [FitzGerald and Coyle, 2006] the model was extended to stereo and multi-channel signals forming the convolutive non-negative CP model given by

$$\mathcal{X}_{i,j,k} \approx \sum_{\tau,d} \boldsymbol{A}^{\tau}_{i,d} \boldsymbol{B}_{j-\tau,d} \boldsymbol{C}_{k,d}.$$

In [Schmidt and Mørup, 2006; Mørup et al., 2008e] the cNMF model was generalized to form a 2D convolution as well as extended to the CP model forming a convolutive CP (cCP) based on the model

$$\mathcal{X}_{i,j,k} \approx \sum_{\tau,\phi,d} \mathcal{A}_{\phi,\tau,d}\mathcal{B}_{i-\phi,j-\tau,d}\boldsymbol{C}_{k,d}.$$

Also this model was estimated using multiplicative updates. To handle the overcompleteness of the representation when including many lags τ and ϕ sparseness was imposed on one of the modes. The model was used in [Schmidt and Mørup, 2006; Mørup et al., 2008e] to analyze image and sound data, see Figure 4.8.

4.2.3 Convolutive Sparse Coding

Convolutive Sparse Coding (cSC) is equivalent to the convolutive ICA model but rather than imposing independence, $\boldsymbol{S}$ is constrained to be sparse. Thus, the convolutive Sparse Coding (cSC) model reads

$$\boldsymbol{X}_{i,j} \approx \sum_{\tau,d} \boldsymbol{A}_{i,d}^{\tau}\boldsymbol{S}_{d,j-\tau}, \quad \boldsymbol{x}_j \approx \sum_{\tau} \boldsymbol{A}^{\tau}\boldsymbol{s}_{j-\tau}.$$

where $\boldsymbol{S}$ is sparse. This model was originally motivated to form representations that were invariant under temporal shifts, i.e. if the filter $\boldsymbol{A}^{\tau}$ constitutes a signal across time τ then the sparse code $\boldsymbol{S}$ will denote the onset of this signal and as such can code for temporal shifts in these feature signals, see also [Lewicki and Sejnowski, 1999; Blumensath and Davies, 2004b; Smith and Lewicki, 2005; Blumensath and Davies, 2004a]. In [Olshausen, 2003] the model was generalized to code video images.

In [Madsen et al., 2008] we explored that the model is separable in the frequency domain and can be optimized using the following objective of the form given in (4.1)

$$C = \frac{1}{2J}\sum_{f} \|\widetilde{\mathbf{x}}_f - \widetilde{\mathbf{A}}_f\widetilde{\mathbf{s}}_f\|_F^2 - \lambda\sum_{j} log(sp(\mathbf{s}_j)).$$

Where the first term is the reconstruction error and second term the sparsity penalty imposed with strength λ given by the sparse prior distribution sp. We considered the Laplace prior given by $sp(\mathbf{s}_j) \propto e^{-\|\mathbf{s}_j\|_1}$ forming a l_1-norm regularization penalty. The sparsity in the time domain as well as regions where the filter $\mathbf{A}^{\tau}$ is zero is not transparent in a frequency domain representation. However, the convolutive model is efficiently estimated in a frequency domain representation. Thus, again the TFGM admits the benefits of the representations in the two domains. The gradient of the least squares error in the frequency

domain is given by

$$\begin{aligned}\nabla_{\widetilde{\boldsymbol{A}}_{i,d,f}} C &= -\frac{1}{J}(\widetilde{x}_{i,f} - \sum_d \widetilde{a}^*_{i,d,f}\widetilde{s}_{d,f})\widetilde{s}^*_{d,f} \\ \nabla_{\widetilde{\boldsymbol{S}}_{d,f}} C^{lS} &= -\frac{1}{J}\sum_i \widetilde{x}_{i,d,f}(\widetilde{x}_{i,f} - \sum_d \widetilde{a}^*_{i,d,f}\widetilde{s}_{d,f})\end{aligned}$$

Thus, the gradient in the time domain is given by

$$\begin{aligned}\nabla_{\boldsymbol{A}^\tau} C &= \mathcal{F}^{-1}(\nabla_{\widetilde{\mathbf{A}}} C)_\tau \\ \nabla_{\boldsymbol{S}} C &= \mathcal{F}^{-1}(\nabla_{\widetilde{\mathbf{S}}} C^{LS}) + \lambda \operatorname{sign}(\boldsymbol{S})\end{aligned}$$

Hence, by computing the gradient in the time domain it becomes transparent how $\boldsymbol{A}^\tau$ can be estimated such that only active regions of the filter $\boldsymbol{A}^\tau$ are updated. Notice, how the complexity of this approach does not increase when increasing the filter length. Furthermore, the update in the time domain of $\boldsymbol{S}$ enables the combination of sparseness constraint in the time domain with efficient representation in the frequency domain. While $\boldsymbol{A}^\tau$ can be updated using the normalization invariance approach described in section A.1.3.2, $\boldsymbol{S}$ can be updated using the GB-SC algorithm described in section 2.5. In Figure 4.9 we demonstrate the cSC algorithm on the EEG-data set described in [Makeig et al., 1999] based on a visual paradigm. We removed the three frontal electrodes EOG1, EOG2 and FPz heavily confounded by eye artifacts prior to the analysis.

4.3 Rotation and Shift Invariant Sparse coding

While the previous sections considered various types of shift in the data we will here consider more general transformation of the components. The approach will be motivated by the representation of shifts and rotations of objects in the processing of information in visual area 1 (V1) of the human brain, see also [Mørup and Schmidt, 2008].

When we experience our surroundings, it is well known that our perception does not alter when we move the head or change gaze. Thus, our brain manages to interpret the world, despite the location and orientation of the objects we navigate among. The visual area 1 (V1) in the visual cortex of the human brain is retinotopically organized, such that neighboring regions of the retina are also neighboring regions in V1 [Tootell et al., 1982]; however, the visual processing in the human brain is also organized into orientation selective columns [Albright, 1984; Hubel, 1995]. Here, a typical simple cell (typical simple cell receptive fields are illustrated in Figure 4.10) responds best to some optimum

stimulus orientation, and the response measured by the number of impulses, as the receptive field is passed through, falls off over 10–20 degrees to either side of the optimum, declining steeply to zero outside this region. If an electrode is pushed through the cortex in a direction parallel to the surface, an amazingly regular sequence of changes in orientation occurs. Every time the electrode advances 0.05 millimeter, the preferred orientation shifts on average about 10 degrees clockwise or counterclockwise [Hubel, 1995]. Thus, it would be desirable to take into account this highly organized structure of the visual processing in the brain, both in terms of retinotopy as well as orientation selective columns for the formation of rotation and shift invariant representations.

It is demonstrated in [Olshausen and Field, 1996] how Sparse Coding of image patches results in Gabor like features, see also Figure 2.5, based on the following model (equivalent to the factor analysis model) [Olshausen and Field, 1996, 1997]

$$I^{(k)}(x,y) \approx \sum_{d=1}^{D} \alpha_{k,d} \Psi_d(x,y).$$

where, $I^{(k)}(x,y)$ denotes the kth image patch of the same size as the desired feature images, $\Psi_d(x,y)$, and α_d is the sparse code. Hence, image patches are approximated by $R^{(k)}(x,y) = \sum_{d=1}^{D} \alpha_{k,d} \Psi_d(x,y)$, i.e., a sparse linear combination of the feature images. Thus, the model is equivalent to the Sparse Coding model given in (2.6) thus can be estimated by

$$\arg\min_{\alpha,\Psi} \sum_{k=1}^{K} (\mathcal{D}(\boldsymbol{I}^{(k)}, \boldsymbol{R}^{(k)}) - \lambda \sum_{d=1}^{D} \log sp(\alpha_{k,d}))$$

where λ is a parameter, that defines the tradeoff between reconstruction error and sparseness of the code. $\mathcal{D}(\cdot,\cdot)$ is a distance measure of the reconstruction error, for example the least squares error, and sp is again the sparse prior distribution of $\alpha_{k,d}$, for example given by the Laplace distribution $sp(\alpha_{k,d}) \propto e^{-|\alpha_{k,d}|}$.

In [Eggert et al., 2004; Wersing et al., 2003] it is demonstrated how the Sparse Coding model can be extended to general transformation invariance of the feature images. The features are here invariant to a pre-specified set of operators, T_r

$$I^{k}(x,y) \approx \sum_{d=1}^{D} \sum_{r=1}^{R} \alpha_{k,d,r} T_r(\Psi_d)(x,y).$$

These operators, T_r, account for any desired transformation within each patch, such as scaling and rotation. The model is based on subdividing the image into image patches; thus, a drawback of the above approach is that the extracted features depend on how the image is subdivided, and the model cannot account for simple transformations such as shifts without introducing redundant features.

In [Mørup and Schmidt, 2008] we proposed the following model, that does not rely on subdividing the image into patches, and allows the features to be invariant to a given set of pre-specified transformations, T_r. Let $I \in \mathbb{R}^{X \times Y}$ be the full image (without subdividing), then

$$\boldsymbol{I} \approx \sum_{d=1}^{D} \sum_{r=1}^{R} \boldsymbol{\alpha}_{d,r} * T_r(\boldsymbol{\Psi}_d).$$

where $*$ denotes 2-dimensional convolution in valid regions, $\boldsymbol{\alpha}_{d,r} \in \mathbb{R}^{(X+U-1)\times(Y+V-1)}$ and $\Psi_d \in \mathbb{R}^{U \times V}$. The above model is related to the convolutive Sparse Coding model of the previous section, see also [Schmidt and Mørup, 2006; Mørup et al., 2008e; Lewicki and Sejnowski, 1999; Lee et al., 2007; Blumensath and Davies, 2004a], with the extension of invariance to general transformations. The proposed model directly implements shift invariance through 2-D matrix convolution, which can be efficiently implemented in the Fourier domain. In the following, in addition to shift invariance, we consider invariance to rotation. Thus, T_r denotes a rotation operator, such that $T_r(\Psi_d)$ rotates the feature image, Ψ_d, $2\pi(r-1)/R$ radians clockwise. From this formulation of shift and rotation invariant Sparse Coding, a strong resemblance can be found between each component of the sparse code and the retinotopic organization in the human brain subdivided into orientation selective columns (see Figure 4.11). However, contrary to the organization in the brain the above model explicitly enforces the extracted features to be consistent across the visual field. No such explicit constraint is given for the processing of visual information in the brain.

To incorporate both shift and rotation invariance with respect to R different rotations, the sparse code, $\alpha \in \mathbb{R}^{(X+U-1)\times(Y+V-1)\times R \times D}$, becomes huge compared to the previous shift and rotation dependent image decompositions, in which the analyzed images were subdivided into image patches prior to the analysis, resulting in a sparse code of size $\boldsymbol{\alpha} \in \mathbb{R}^{K \times D}$. Thus, it is not feasible to solve for the sparse code using traditional Sparse Coding algorithms based on computing the Hessian matrix. Thus, in order to estimate the parameters of the model, we used the memory efficient GB-SC algorithm derived in [Mørup and Schmidt, 2008], see Algorithm 2.5c.

As we would like the model to extract features that are similar across various different images, we extended the model to N images of arbitrary size

$$\boldsymbol{I}^{(n)} \approx \boldsymbol{R}^{(n)} = \sum_{d=1}^{D} \sum_{r=1}^{R} \boldsymbol{\alpha}_{d,r}^{(n)} * T_r(\boldsymbol{\Psi}_d).$$

Hence, the nth image is modeled by a sparse code, $\boldsymbol{\alpha}_{d,r}^{(n)}$, convolved with the pre-specified transformations of set of feature images, $T_r(\boldsymbol{\Psi}_d)$, that are shared by all the N images.

Using the least squares error for the reconstruction penalty and the Laplace prior for the sparse code, we obtain the following objective

$$\sum_{n=1}^{N} \frac{1}{2} \|\boldsymbol{I}^{(n)} - \boldsymbol{R}^{(n)}\|_F^2 + \lambda \sum_{d,r} \|\boldsymbol{\alpha}_{d,r}^{(n)}\|_1 \tag{4.11}$$

Presently, we consider rotation invariant features, thus, r indexes a set of predefined rotation operators. For an illustration of this, see Figure 4.11 and 4.12.

The derivative of the objective function (4.11) with respect to $\boldsymbol{\alpha}_{d,r}^{(n)}$ and Ψ_d is

$$\nabla \boldsymbol{\alpha}_{d,r}^{(n)} = \left(\boldsymbol{R}^{(n)} - \boldsymbol{I}^{(n)}\right) * T_\pi(\boldsymbol{\Psi}_d) + \lambda \operatorname{sign}\left(\boldsymbol{\alpha}_{d,r}^{(n)}\right), \tag{4.12}$$

$$\nabla \Psi_d = \sum_{n=1}^{N} \sum_{r=1}^{R} T_r^{-1}\left(\boldsymbol{R}^{(n)} - \boldsymbol{I}^{(n)}\right) * T_r^{-1}\left(T_\pi\left(\boldsymbol{\alpha}_{d,r}^{(n)}\right)\right), \tag{4.13}$$

where T_r^{-1} denotes the inverse rotation operator, and T_π denotes rotation of 180 degrees. We implemented the rotation operator T_r using linear interpolation between the image pixels. In image regions, where T_r^{-1} and T_r are not valid, we zero padded the data. $\boldsymbol{\Psi}_d$ was updated such that $\|\boldsymbol{\Psi}_d\|_F = 1$ using the normalization invariance approach described in section A.1.3.2.

Figure 4.12 shows the result of a rotation and shift invariant sparse coding of a synthetically generated dataset. The data consists of a number of bar and C-shapes, randomly rotated between 20 uniformly distributed orientations over the interval $[0; 360°[$.

Figure 4.13 shows the results analyzing the natural images used in [Olshausen and Field, 1996] based on shift-invariance and rotation-and-shift-invariance. Whereas shift invariance alleviates redundancies in terms of shifts including rotation invariance further removes redundancy to rotation. As a result, the features extracted including rotation invariance more closely resemble simple cell receptive fields than the Gabor-like features of Sparse Coding and shift invariant sparse coding. In the shift invariant model, we used 10 features with feature image size 16×16. In the rotation and shift invariant model, we used 4 feature images of size 20×20 and 10 rotational representations covering the interval $[0; 180°[$ (due to rotation of the features the valid regions was comparable to features of size 16×16). The data set taken from [Olshausen and Field, 1996] consists of 10 natural images of size 512×512. Thus, the resulting sizes of the sparse codes were $(512+16-1) \times (512+16-1) \times 10 \times 1 \times 10 = 27,772,900$ variables and $(512+20-1) \times (512+20-1) \times 10 \times 10 \times 4 = 112,784,400$ variables respectively.

4.4 Summary

This chapter considered extensions of the factor analysis based decomposition to handle delays as well as reverberation and general transformation of the sources. Often the factor analysis based decomposition is too restricted, thus, it is relevant to consider various extensions. As such we saw how the component identification improved when incorporating delays in the CP model and also how features that more closely resemble simple cell properties could be obtained when incorporating invariance to shift and rotation in the sparse coding of natural images. Presently we considered only a very limited class of invariance, i.e., shift, reverberation and rotation. However, changes such as time warping and more general non-linear transformations also form important extensions to the factor analysis based decompositions [Tomasi, 2006].

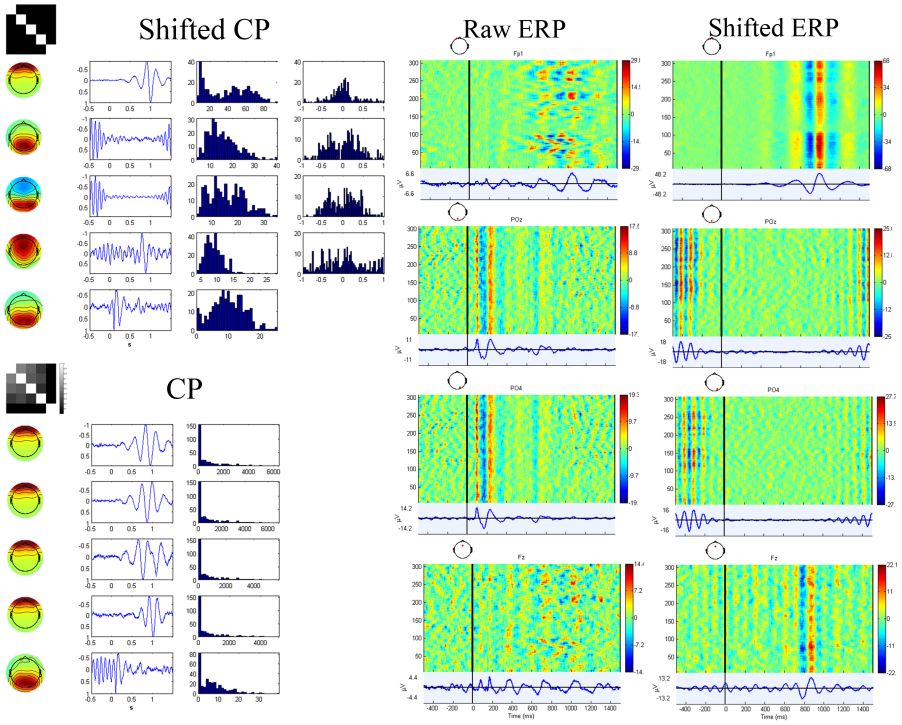


Figure 4.7: A 5 component ShiftCP analysis (top left panel) of a 64 channel x 1024 time-points x 313 trials event related EEG data set described in [Mørup et al., 2008a] as well as the corresponding regular instantaneous CP analysis (bottom left panel). For each component is given the spatial map $\mathbf{a}_d$, time series $\mathbf{b}_d$, histogram of trial strengths $\mathbf{c}_d$ and histogram of delays $\boldsymbol{\tau}_d$. Above the decomposition is given the absolute correlation between the various components. Clearly, the instantaneous CP model has found a degenerate solution in which the activity of the eye-blink has been captured in the four first components. Thus whereas the correlation between the factors is very small such that each component model different aspects of the data in the ShiftCP model a degenerate solution is obtained in the regular CP analysis (see correlation matrices in the top left corner of each decomposition). To the right is given the ERP image and event related potential for the channel having the maximal activity in each of the four shifted components as well as the activity when shifting the EEG data according to the estimated component delays (the ERP images are smoothed with a Gaussian window $\sigma = 10$). Whereas the ShiftCP model accounts for 36 % of the variance the instantaneous CP model only accounts for 21 % of the variation in the data. Thus, extending the CP model to incorporate shifts greatly improves component identification.

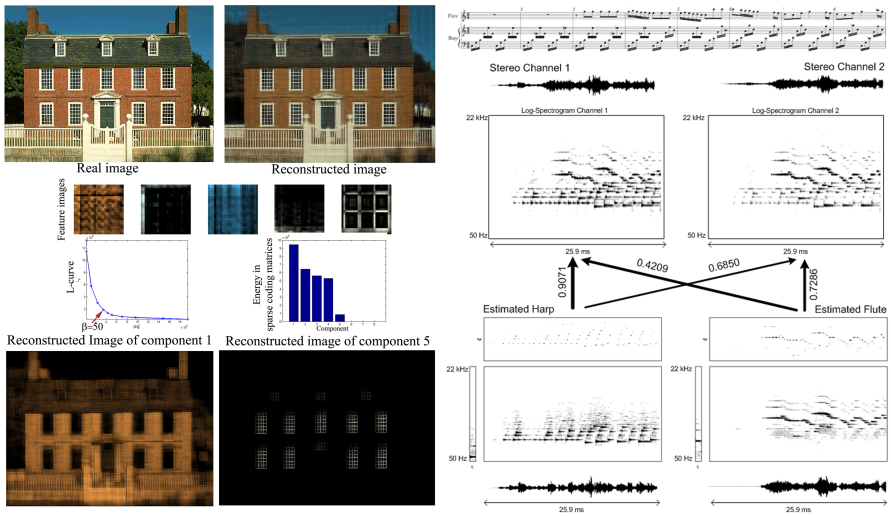


Figure 4.8: **Left panel:** A 2D convolutive CP analysis of an image of a brick house. To the top left is given the raw image and to the top right the image reconstructed from the feature images given below. The degree of regularization is given by evaluating the tradeoff between reconstruction and regularization while the energy of each of the 8 features are given in the bar plot. At the bottom is given the reconstructed image from component one and 5 mainly pertaining to the bricks and window frames of the house respectively. **Right panel:** A 2D convolutive CP analysis of a stereo recording of the "Fog is Lifting" by Carl Nielsen [Jensen and Johansen, 19xx]. To the top the score is given and just below the waveform of the two stereo channels as well as the amplitude of the log-spectrogram of the signal over time for the two channels. The 2D convolutive CP analysis has well separated the stereo music into the harmonical structure of the two instruments of the music, i.e. the harp and flute respectively and the scores the instruments play. By convolving the estimated harmonic structure of the instruments with the estimated scores result in an estimate of the log-spectrogram of the sound produced by each instrument. Arrows indicate the estimated mixing strength of the harp and flute respectively into the two stereo channels. The waveform of each instrument is given at the bottom using spectral masking based on the log-spectrogram of the two instruments found. For more details on the above decompositions of music and image data consult [Mørup et al., 2008e].

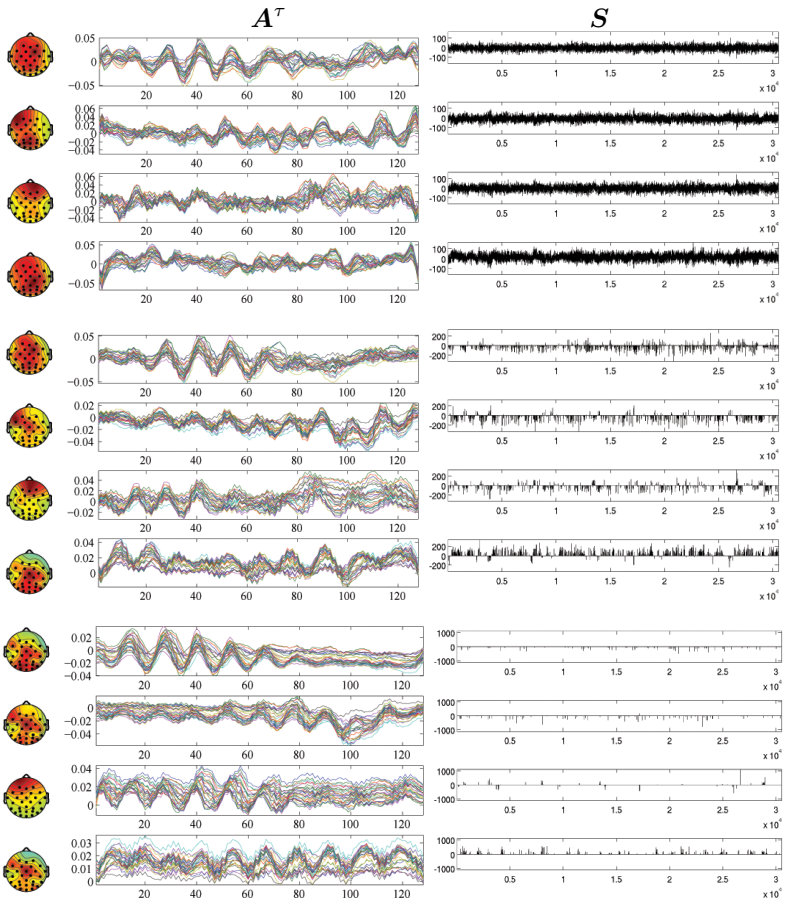


Figure 4.9: Convolutive Sparse Coding analysis of EEG data obtained from a visual paradigm sampled at 128 Hz. The size of the data is $\boldsymbol{X} \in \mathbb{R}^{29\times 30504}$ while $\boldsymbol{A}^\tau \in \mathbb{R}^{29\times 4}$ and $\tau \in [1, 2, \ldots, 128]$, i.e. the filter length is 1 second. **Top panel:** Analysis for $\lambda = 0$, clearly $\boldsymbol{S}$ given to the right is not sparse thus the EEG activity is modeled both in the convolutive filter $\boldsymbol{A}^\tau$ and in the sources $\boldsymbol{S}$. The scalp maps to the left gives the power of the filter coefficients for the electrodes of each component. The explained variation is 91% **Middle panel:** Analysis based on $\lambda = 200$, clearly $\boldsymbol{S}$ has become sparse while the temporal structure of the EEG data mainly is coded in the filter $\boldsymbol{A}^\tau$. The explained variation is 66%. **Bottom panel:** When increasing the sparsity strength ($\lambda = 700$) $\boldsymbol{S}$ becomes even more sparse. The explained variation is 35%. The activity captured by the models are mainly the powerful alpha activity residing in a frequency band around 8-12 Hz.

Figure 4.10: Typical receptive-field maps for V1 simple cells [Hubel, 1995]. The off-regions and on-regions of the cells are illustrated by the black and white colors.

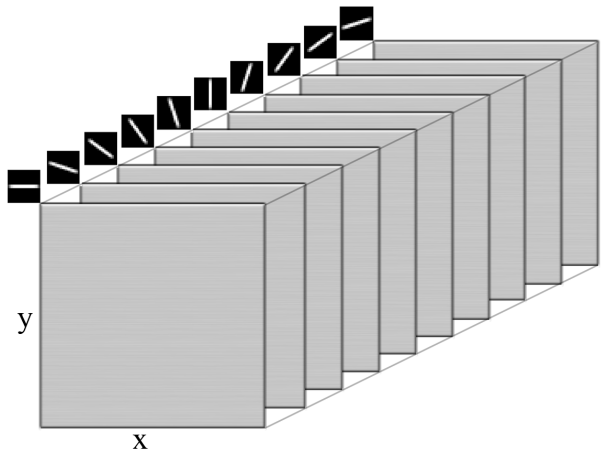


Figure 4.11: Illustration of the Sparse Coding array for a given component, d, i.e. $\boldsymbol{\alpha}_{d,r}$ for $r \in \{1, 2, \ldots, 10\}$ of a total of R=10 rotations corresponding to the feature image of a bar being represented in the interval $[0°; 180°[$. The sparse code representation is similar to the organization of V1 of the human visual cortex, where the organization of the cells maintain the organization of the receptive field of the eye, i.e. the x and y coordinates, while each receptive field in V1 is organized into orientation selective columns, corresponding here to the indexing r over rotations.

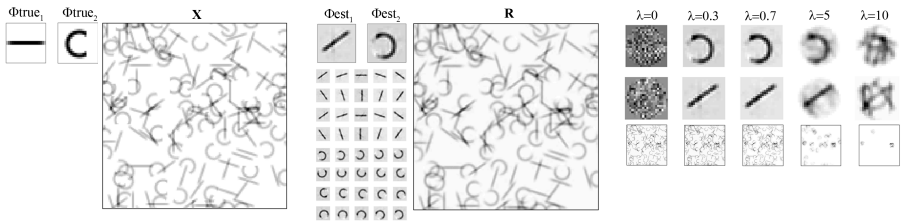


Figure 4.12: A rotation and shift invariant sparse coding analysis of a synthetically generated dataset. **Left panel:** The feature images consist of a bar and a C shape, randomly rotated to generate the synthetic image data $\boldsymbol{X}$. **Middle panel:** Estimated features and data using the rotation and shift invariant sparse coding algorithm. **Right panel:** Inspection of the results obtained for different values of the regularization parameter λ. Note that the gray background of the features are due to a different color axis used to show small regions of the estimated feature images with negative values. When the regularization strength is week, most of the information is coded in the sparse code, while for the "correct" degree of sparsity, the information of the bar and C-shape is coded in the features. When the regularization is too strong, only the most prominent regions are coded, which results in features that are highly localized.

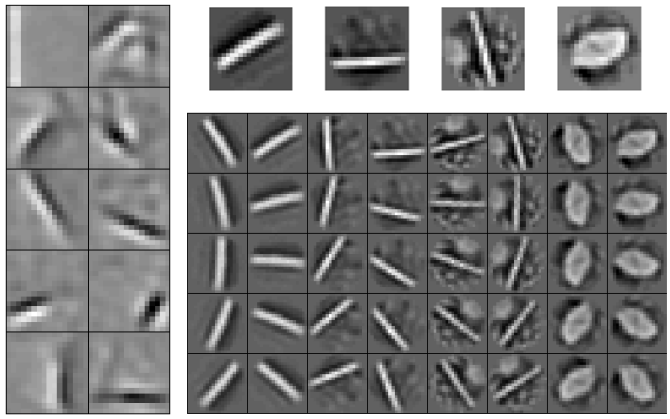


Figure 4.13: **Left panel:** Shift invariant feature images, $\mathbf{\Psi}$, of size 16×16 obtained when analyzing the natural image data using shift invariant sparse coding. Similar to Sparse Coding, Gabor-like features are obtained; however, the features are not redundant with respect to shift, since the model can use each feature at any position. The features appear, however, are redundant with respect to rotation. **Right panel:** Rotation and shift invariant feature images $\mathbf{\Psi}$ of size 20×20 obtained when analyzing the natural image data using the rotation and shift invariant sparse coding algorithm. Notice, due to the rotation invariance, only the central areas of the features are non-zero. The first feature obtained seem to mimic on-center off-surround behavior coding for luminance, while the remaining three features resembles edge detectors varying from high-frequent to low frequent edges. The features obtained seem to closely resemble the typical simple cell behavior given in Figure 4.10. To the right, the 10 rotated representations of the four features are shown.

Chapter 5

Decomposition Methods for Clustering

The term *cluster analysis* was first used in 1939 by Tryon [Tryon, 1939]. Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification, i.e., it models data by its clusters [Berkhin, 2002]. Clustering is a central problem in Unsupervised Learning. While most work has focused on clustering of data vectors in $\mathbb{R}^n$, clustering of graph-structured data, also known as *graph partitioning* or *community detection*, has become increasingly important with the growing interest in understanding data coming from biological, communication, computer, and social networks. The literature on clustering is extensive. A review of algorithms for point clustering can be found in [Berkhin, 2002] while a good introduction to the structure of complex network can be found in [Newman, 2003]. A description of methods for community detection in complex networks can be found in [Newman, 2006a]. A recent tutorial on spectral clustering can be found in [von Luxburg, 2007]. Presently, we will focus on the connection between clustering and matrix decomposition with emphasis on the results given in [Mørup et al., 2008c].

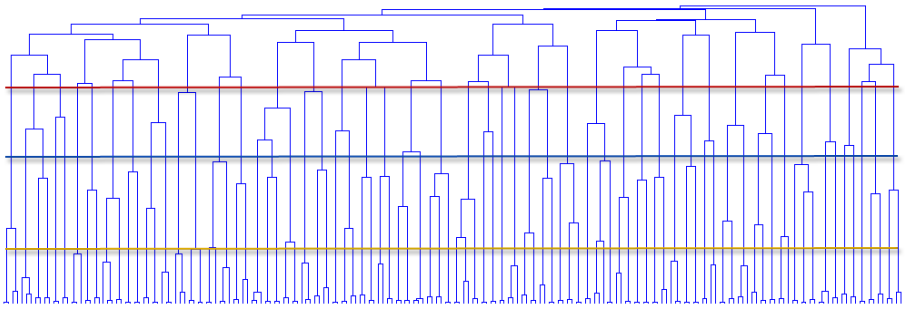


Figure 5.1: Hierarchical clustering represents the data points in a tree also called a dendrogram. Considering given levels of the tree given by the red, blue and yellow lines form a partitional clustering of the data.

5.1 Point Clustering

Point clustering is most often referred to as geometric clustering or k-means clustering, however, the latter refers to the well-known iterative refinement algorithm. The point clustering problem can be separated into two approaches; hierarchical algorithms that successively finds clusters and partitional algorithms that determine all clusters at once.

5.1.1 Hierarchical Clustering

Hierarchical clustering represents the data points in a tree also called a dendrogram with individual elements at one end united into a single cluster at the other end. Naturally this gives rise to a top down or bottom up approach, i.e.

- Agglomerative: Bottom up approach in which nodes or groups of nodes are joined based on some form of distance metric between data points.
- Divisive: Top down approach in which clusters are recursively split into smaller clusters.

While each level of the hierarchical clustering problem can be found using a partitional clustering approach, see Figure 5.1, the hierarchical clustering problem has no direct matrix decomposition formulation forming the full dendrogram. Thus, the focus will here be on partitional clustering that admit a problem formulation in terms of matrix decomposition.

5.1.2 Partitional Clustering

Perhaps the most well known algorithm for partitional clustering is the k-means algorithm proposed in [Hartigan, 1975]. The goal is to assign n points in $\mathbb{R}^m$ to k clusters with cluster centroids $\mathbf{c}_1, \ldots, \mathbf{c}_k$. This may be formulated as a discrete optimization problem: the k centroids must be chosen so that the distance between the data points and their assigned cluster centroids, i.e. the quantity $\sum_{\ell=1}^{k} \sum_{\mathbf{x}_j \in P_\ell} \|\mathbf{x}_j - \mathbf{c}_\ell\|_2^2$, is minimized. Here P_ℓ denotes the set of nodes belonging to cluster ℓ. A naive solution involves an exhaustive search over all possible choices of P_ℓ and $\mathbf{c}_\ell$. In fact, the problem is known to be NP-hard even when $k = 2$. The k-means algorithm is based on a two step procedure - a) assign data to clusters, b) estimate new cluster centroids. As such the algorithm is strongly related to the two step Expectation Maximization (EM) procedure [Moon, 1996] where the E-step estimates probabilities, which is equivalent to a soft (fuzzy) reassignment and the M-step finds an approximation to a mixture model, given current soft assignments [Berkhin, 2002; Banerjee et al., 2005]. As demonstrated in [Banerjee et al., 2005] this approach can easily be generalized to other distance metrics than least squares. As such, measuring distances by the l_1-norm is also referred to as k-medians as the cluster centers by this metric are given by the median point of the data belonging to the given cluster.

5.1.3 Point Clustering as a Matrix Decomposition Problem

The k-means clustering problem can in matrix notation be written as

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{S}\|_F^2 \\ \text{subject to} & \|\mathbf{s}_j\|_0 = 1, \quad \mathbf{s}_j \in \{0,1\} \end{array} \tag{5.1}$$

where $\boldsymbol{X}, \boldsymbol{C}$, and $\boldsymbol{S}$ are real-valued matrices of sizes $m \times n$, $m \times k$, and $k \times n$ respectively. The discrete combinatorial nature of the problem is hidden in the fact that the constraints $\|\mathbf{s}_j\|_0 = 1$ and $\mathbf{s}_j \in \{0,1\}$ when imposed together forces $\boldsymbol{S}$ to be a $\{0,1\}$-valued *indicator matrix* — $s_{\ell j}$ indicates whether the jth observation belongs to cluster ℓ ($s_{\ell j} = 1$) or not ($s_{\ell j} = 0$).

5.1.4 Spectral Relaxations

Spectral relaxations of clustering problems formulated in terms of matrix decompositions/approximations such as principal component analysis (PCA) and nonnegative matrix factorization (NMF) or their variants, have lately attracted considerable interest [Ng et al., 2001; Zha et al., 2001; Ding et al., 2005, 2006b;

von Luxburg, 2007] as computationally tractable approximations to the original NP-hard problems. Spectral clusterings essentially relax the *discrete* optimization problems of clustering to *continuous* optimization problems whose objectives share common properties. In [Mørup et al., 2008c] we demonstrated how (5.1) admits the following l_1-relaxation:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{X} - \boldsymbol{C}\boldsymbol{S}\|_F^2 \\ \text{subject to} & \|\mathbf{s}_j\|_1 = 1, \quad 0 \leq \mathbf{s}_j. \end{array} \tag{5.2}$$

Hence, $\boldsymbol{S}$ is no longer required to be $\{0,1\}$-valued. The constraints $\|\mathbf{s}_j\|_1 = 1$, $0 \leq \mathbf{s}_j$ say that $\mathbf{s}_j$ is a point on the unit simplex thus the l_1 relaxed k-means problem we denoted *Simplicial Point Clustering* (SPC), being a simplicial relaxation of the original k-means point clustering problem. In fact, one may interpret $s_{\ell j}$ to be the probability that the jth data point belongs to the cluster with center $\mathbf{c}_\ell$. The l_1-relaxation is naturally motivated by the fact that $\|\mathbf{s}\|_1$ is the largest convex underestimator of $\|\mathbf{s}\|_0$ on $\{\mathbf{s} \mid \|\mathbf{s}\|_\infty \leq 1\}$. An l_2-relaxation, on the other hand [Ng et al., 2001; Bach and Jordan, 2004], cannot be similarly motivated. For SVD /PCA, one minimizes $\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\|_F^2$ subjected to the constraints $\|\mathbf{u}_i\|_2 = \|\mathbf{v}_j\|_2 = 1$ and $\boldsymbol{\Sigma}$ being a diagonal matrix, see also section 2.1. One can prove that the optimal solution is given by an orthogonal matrix $\boldsymbol{U}$ representing the most dominant subspace of the data. For NMF [Ding et al., 2005; Have et al., 2006], one minimizes $\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2$ subjected to the constraints $\boldsymbol{W} \geq 0$ and $\boldsymbol{H} \geq 0$ such that the data points (with appropriate normalization of the components) are spanned by the convex hull of $[\mathbf{0}\ \boldsymbol{W}]$ as described in section 2.4. While the above SPC method similarly projects data points to the convex hull of $\boldsymbol{C}$, the entries of $\boldsymbol{C}$ in (5.2) may well be negative. In Figure 5.2, the qualitative difference between the spectral relaxations of k-means given by SVD/PCA, NMF and SPC are illustrated. In Figure 5.3 is given a SPC analysis of the `www.netflixprize.com` data.

5.2 Graph Clustering

Let $G = (V, E)$ be a graph with vertex set $V = \{1, \ldots, n\}$. Let $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ be the adjacency matrix of G, where we will always assume that $x_{ii} = 1$, i.e. every node is connected to itself. The graph G is called *undirected* if $m = n$ and $x_{ij} = x_{ji}$ and *directed* otherwise. Furthermore, the graph is called *unweighted* if $x_{ij} \in \{0, 1\}$ and *weighted* otherwise. In many applications involving networks and graph structured data, it is critical to cluster nodes into highly interconnected units with low intra-cluster connectivity. This is referred to variously as community detection, graph partitioning, or graph clustering. As in the case of point clustering, the problem is NP-hard.

5.2.1 Graph Clustering as a Matrix Decomposition Problem

In [Mørup et al., 2008c] we proposed the metric

$$D_\rho(\boldsymbol{X} \parallel \boldsymbol{A}) := (1-\rho) \sum_{x_{ij} \neq 0} |x_{ij} - a_{ij}|^2 + \rho \sum_{x_{ij}=0} |x_{ij} - a_{ij}|^2. \tag{5.3}$$

Using this metric the graph clustering problem may be recast as the following matrix approximation problem with l_0-norm constraints:

$$\begin{array}{rl} \text{minimize} & D_\rho(\boldsymbol{X} \parallel \boldsymbol{R}^\top \boldsymbol{S}) \\ \text{subject to} & \|\mathbf{s}_j\|_0 = 1, \quad \mathbf{s}_j \in \{0,1\}, \\ & \|\mathbf{r}_i\|_0 = 1, \quad \mathbf{r}_i \in \{0,1\}. \end{array} \tag{5.4}$$

The matrices $\boldsymbol{R}$ and $\boldsymbol{S}$ form the clustering of the nodes in the row and column spaces of $\boldsymbol{X}$ respectively. For undirected graphs $\boldsymbol{R} = \boldsymbol{S}$. Since $x_{ii} = 1$ the objective above favors clustering nodes to themselves. The ability of the above objective to correctly cluster graphs is based on the following theorem derived in [Mørup et al., 2008c]

Theorem 5.1 *Consider an ensemble of undirected graphs with random adjacency matrices $\boldsymbol{X} \in \mathbb{R}^{n \times n}$. For each member of the ensemble define a random cluster assignment and let the adjacency matrix be drawn so that the average link densities are fixed within clusters (ρ_{clust}) and between clusters (ρ_{noise}), finally let $\rho_{clust} > \rho > \rho_{noise}$. Then the ensemble expectation of the cost D_ρ for the 'correct cluster assignment' is lower than the ensemble expectation of the cost for any configuration with a single misplaced node relative to the 'correct' assignment.*

Proof. Consider the sub-ensemble defined by a fixed assignment, i.e., with some k clusters of arbitrary but given sizes n_j such that $\sum_{j=1}^k n_j = n$. Consider a configuration with a node wrongly assigned to cluster w instead of r. The expected value of D_ρ^w for this configuration is

$$\frac{1}{2} E[D_\rho^w] = (1-\rho)\rho_{clust} n_r + \rho(1-\rho_{noise}) n_w + (1-\rho)(\rho_{noise}) \sum\nolimits_{j \neq r,w} n_j + c,$$

where the first term represents the loss from edges *not* accounted for in the true cluster r, the second term represents the cost of predicting links in the wrong places in the wrong cluster w, the third term the cost of neglecting to predict noise links and c is a constant giving the cost of the remaining graph not pertaining to the wrongly assigned node. The expected cost of the correct assignment is

$$\frac{1}{2} E[D_\rho^r] = \rho(1-\rho_{clust}) n_r + (1-\rho)(\rho_{noise}) \sum\nolimits_{j \neq r} n_j + c,$$

where the first term is the cost of predicting absent edges in the correct cluster and the second term is the cost of not predicting noise-induced edges. The difference $\mathsf{E}[D_\rho^w] - \mathsf{E}[D_\rho^r] = (\rho_{clust} - \rho)n_r + (\rho - \rho_{noise})n_w$ is guaranteed to be positive if $\rho_{clust} > \rho > \rho_{noise}$. This holds for any value of the cluster sizes, i.e., as long as ρ is above the noise density and less than the clustering density. Thus the result holds in any sub-ensemble defined by a set of cluster sizes, hence in general.

The condition $\rho_{clust} > \rho > \rho_{noise}$ is critical to the accuracy of the method. If ρ is set too small the objective will favor clustering noise, if ρ is set too high, i.e. above the density of the clusters, the objective is not guaranteed to favor correct classification [Mørup et al., 2008c].

Setting ρ to the average density of the graph agrees with one's natural notion of clusters in a graph, that is, clusters are precisely the regions that are denser than the average density. With this choice of ρ, the quantity D_ρ measures deviation from the 'null hypothesis' of having average density and so clusters that are denser than average will be favored by (5.5) and (5.4). As such, the parameter ρ defines a cluster admissibility criterion, i.e. a threshold defining when nodes are to be potentially clustered together — when their inter-connectivity is above this threshold. .

5.2.2 Simplicial Clustering, Graph Laplacian and Modularity

The l_1-relaxation of (5.4) proposed in [Mørup et al., 2008c] is given by

$$\begin{array}{rl} \text{minimize} & D_\rho(\boldsymbol{X} \parallel \boldsymbol{R}^\top \boldsymbol{S}) \\ \text{subject to} & \|\mathbf{s}_j\|_1 = 1, \quad 0 \leq \mathbf{s}_j \leq 1, \\ & \|\mathbf{r}_i\|_1 = 1, \quad 0 \leq \mathbf{r}_i \leq 1. \end{array} \tag{5.5}$$

It is evident that (5.5) allows for nodes to belong to several clusters with probabilities given by $\boldsymbol{s}_j$ and $\boldsymbol{r}_j$ respectively. In other words, overlapping communities are naturally modelled by (5.5) forming the *Simplicial Graph Clustering* (SGC) in analogy to SPC.

While $\boldsymbol{X}$, $\boldsymbol{S}$ and $\boldsymbol{R}$ are nonnegative matrices, the resemblance to NMF or orthogonal NMF (ONMF) [Ding et al., 2006b] is only superficial. First, the reconstruction objective is different — $D_\rho(\boldsymbol{X} \parallel \boldsymbol{R}^\top \boldsymbol{S})$ instead of $\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}\|_F^2$. Secondly, even an NMF or ONMF solution based on our reconstruction objective would not have the dual simplicial property reflected in the constraints of (5.5), where the columns and rows of $\boldsymbol{X}$ are simultaneously approximated by convex combinations of points on the unit simplex given by the columns of $\boldsymbol{R}$

and $\boldsymbol{S}$.

The two most popular existing methods for graph clustering are not NMF-based but *spectral partitioning* and *modularity clustering*. Given an undirected graph $G = (V, E)$ with $|V| = n$, $|E| = r$, and adjacency matrix $\boldsymbol{X} \in \mathbb{R}^{n \times n}$. Let $k_j := \sum_{i=1}^{n} x_{ij}$ so that $\mathbf{k}$ is a vector containing the total number of links for each node. The eigenvalue decomposition of the graph Laplacian $\boldsymbol{L} = \text{diag}(\mathbf{k}) - \boldsymbol{X}$ has traditionally been used for spectral partitioning [Fiedler, 1973; Pothen. et al., 1990; von Luxburg, 2007]. A more recent approach proposes to detect community structures in complex networks by maximizing the modularity of the graph [Newman and Girvan, 2004; Newman, 2006b], defined as

$$Q(\boldsymbol{S}) := \frac{1}{4r} \text{tr}(\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^\top), \quad \boldsymbol{B} = \boldsymbol{X} - \frac{1}{2r}\mathbf{k}\mathbf{k}^\top.$$

Here $\boldsymbol{S}$ is a binary matrix with one non-zero entry in each column and $2r = \sum_{j=1}^{n} k_j$. Note that $Q(\boldsymbol{S})$ measures the deviation of the fraction of edges within communities from the expected fraction of such edges. Hence, the model has a built-in null hypothesis that communities should link more than expected in terms of degree distributions. So maximizing the modularity should plausibly give the optimal number of communities. A variety of approaches have been derived to solve for the modularity based on spectral relaxations as well as Gibbs sampling and simulated annealing [Lehmann and Hansen, 2007]. Nevertheless, neither spectral partitioning nor modularity clustering can satisfactorily resolve the following difficulties:

1. What is the right number of clusters?
2. How can overlapping communities be modeled?
3. Are there good convex approximations?

However, the Simplicial Graph Clustering (SGC) addresses all these issues. First of all, the number of components automatically drops out of the SGC if initiated with a k larger than the 'true' number of clusters. This is due to the value of ρ working as a null-hypothesis for the cluster admissibility. Secondly, with the constraints $\|\mathbf{s}_j\|_1 = 1$ and $\mathbf{s}_j \geq 0$, SGC naturally allows for observations to belong to multiple clusters with $\mathbf{s}_j$ being the probabilities of cluster membership. Thirdly, as demonstrated in [Mørup et al., 2008c] the SGC may be solved using classical algorithms for non-negative quadratic programming. As an additional advantage, the SGC solution facilitates easy visualization via the following procedure. The jth node is assigned the cluster with the highest probability $s_{\ell j}$. Clusters are sorted in descending order according to cluster-size and the nodes

within each cluster are in turn sorted in descending order of their probabilities of belonging to that cluster. Thus, the adjacency matrix $\boldsymbol{X}$ is permuted such that largest clusters are in the upper left corner and smallest clusters in the lower right. Within each cluster, the upper left corner corresponds to nodes mainly connected to nodes within the cluster while the lower right corner corresponds to nodes having a higher degree of connectivity to other clusters. Note that the modularity approach, for example, would not work with such visualization methods since its solution $\boldsymbol{S}$ is binary and lacks the information for sorting nodes within clusters.

In [Mørup et al., 2008c] two algorithms were derived to solve for the SPC and SGC respectively based on a modified version of the NNQP given in Algortihm 2.4. For details on how to solve for simplicial constraints see also Appendix A.1.3.3. It is generally difficult to directly compare clustering algorithms since there is no universal yardstick. Even on synthetic data, performance of different methods will depend on underlying assumptions made in the generation of the networks. Thus it is not uncommon to find papers on clustering that do not make their case via comparison with other existing methods but instead rely on applications of the proposed method to real and synthetic networks for evaluating effectiveness. For example, in Handcock et al. [2007], the proposed method is illustrated on two social networks. In M. Hofman and Wiggins [2007]; Kemp et al. [2004]; Newman and Girvan [2004], the performance of the algorithms were evaluated on synthetic and real network data. One may argue that Newman [2006b] is an exception since it compares modularity values for different algorithms. However, this clearly favors algorithms based on modularity optimization. Had we evaluated existing clustering algorithms using D_ρ, the results would invariably favor our method too. Thus, as there is no universal yardstick for comparing different clustering algorithms we evaluated the SGC method on the following three benchmark networks. For a qualitative comparison an analysis of the first two networks by the modularity based on the simulated annealing approach described in [Lehmann and Hansen, 2007] are also given.

Dolphin network:
In Figure 5.4, is given an analysis of the dolphin network described in [Lusseau et al., 2003] and also used in [Newman and Girvan, 2004]. The dolphin network is a social network of relationship between dolphins based on two known disjoint communities having six intercommunity edges. As can be seen from Figure 5.4 the SGC algorithm correctly identifies the underlying two communities while the modularity identifies 4 communities with a great degree of overlap between the last three identifed communities.

Zachary karate club:
The karate club network analyzed by Zachary [Zachary, 1977] is widely used as a test example for methods for detecting communities in complex networks [New-

man, 2004; Girvan and Newman, 2002]. The network consists of 34 members of a karate club as nodes with 78 edges representing friendship between members of the club as observed over a period of two years. Due to a disagreement between the club's administrators and the club's instructors, the club later split in two. Thus, the goal is to be able to predict this split. In Figure 5.5 it can be seen that indeed the SGQ has correctly identified the split of the club into two groups constituting the instructors and administrators respectively. The modularity on the other hand has incorrectly identified three clusters with administrators and instructors clustered together.

College football:
The college football network [Girvan and Newman, 2002] represents the game schedule of the 2000 season of Division I of the US college football league. The nodes represent the 115 teams, while the edges represent 613 regular-season games between teams. The teams are divided into conferences of about 8–12 teams each with on average seven intraconference games and four interconference games. Interconference games are not uniformly distributed hence teams that are geographically close to one another are more likely to play one another than teams separated by large geographic distances. As expected, this dataset forms a network with a great degree of overlaps between potential clusters. In Figure 5.6 a SGC analysis is given.

5.3 Summary

In this chapter it was described how both geometric as well as graph clustering problems can be formulated in terms of l_0-norm constrained matrix decomposition problems as demonstrated in [Mørup et al., 2008c]. As the clustering problem corresponds to a l_0-norm constrained problem it can be considered the extreme goal of simplicity – a problem given by the sparsest attainable code still coding the data (i.e. the l_0-norm set to one). Furthermore, this formulation admits a simplicial relaxation formed by relaxing the l_0-norm constraint to a l_1-norm constraint. An extension that naturally allow to handle overlap between the clusters as demonstrated in Figure 5.2, 5.4, 5.5 and 5.6.

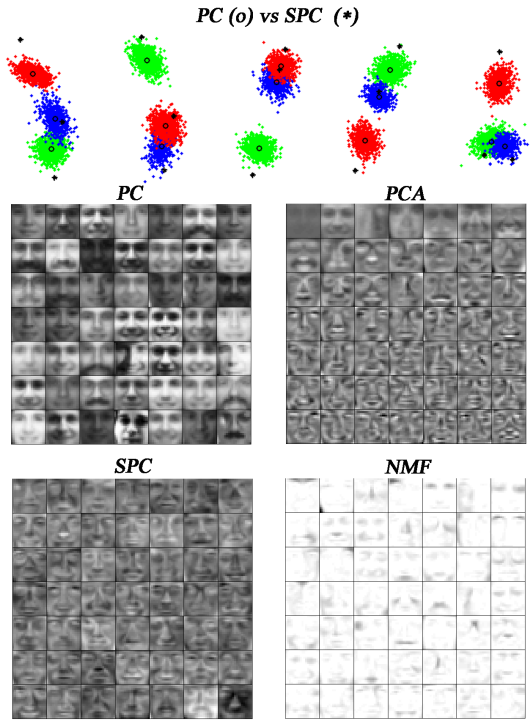


Figure 5.2: **Top panel:** Comparison of the SPC algorithm and regular point clustering (PC) (PC based on the best of 10 k-means) for 5 randomly generated datasets each consisting of 3 clusters. Data resides in 20 dimensional space but projected to the first two principal components for visualization. The estimated cluster centers $\mathbf{c}_\ell$ are marked as black circles (o) and diamonds (*) for the PC and SPC method respectively. Note that the data is according to the SPC algorithm spanned by the convex hull generated by the cluster centers whereas PC gives a hard clustering. Thus, as can be seen from the cluster centers it is favorable to place the cluster centers more to the extreme of the clusters such that the convex hull covers the data points better. Both SPC and PC are within a few percent of the correct classification. **Bottom panel:** Result obtained when clustering the CBCL face data training set of 2429 faces each of 19×19 pixels. Clearly, the extracted cluster centers of the PC correspond more or less to actual faces. The PCA solution on the other hand consist of components ranging from low resolution to high resolution features. Whereas the NMF decomposition gives a part based representation as was also reported in [Lee and Seung, 1999] the SPC algorithm finds a representation resembling more actual faces than the PCA and NMF features. However, the faces obtained looks more like "freaks," i.e. being extreme cases in terms of expression. The explained variation for PC is 96 % while the PCA, SPC and NMF all accounted for more than 99 % of the variation.

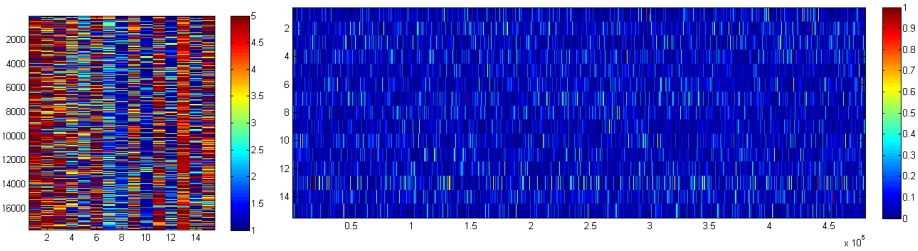


Figure 5.3: SPC analysis of the Netflix data (`www.netflixprize.com`). The Netflix contest is based on a training dataset given by the sparse matrix of 17770 movies by 480189 users with 10^8 ratings taking the values $\{1, 2, 3, 4, 5\}$. The goal is to predict the ratings of movies the users have not yet seen based on the users preferences on rated movies as well as the preferences of other users. This problem is also denoted collaborative filtering. A test set used to evaluate the predictions has been generated by randomly removing some of the ratings from the training data. The first to improve the root mean square error (rmse) of the test set by 10% compared to Netflix own prediction system will win 1 *mio.* US$. We modelled the Movie by Users rating matrix by the model $\boldsymbol{X}^{Movies \times Users} \approx \boldsymbol{W}^{Movies \times 15} \boldsymbol{H}^{15 \times Users}$ using the SPC algorithm. Hence, we modelled the data such that $\|\mathbf{h}_j\|_1 = 1$, $\mathbf{h}_j \geq 0$ while $1 \leq \mathbf{w}_d \leq 5$ such that the columns of $\boldsymbol{W}$ constituted specific user profiles. Entries in $\boldsymbol{X}$ that had not been rated by the users as well as predictions taken out of the data for the test set was treated as missing using the marginalization approach described in Appendix A.1.3.5. To the left is given the estimated user profiles $\boldsymbol{W}$ while to the right is given the probabilities that each user belong to each of the identified user groups (i.e., $\boldsymbol{H}$). While our model had a $rmse = 0.9426$ of the test set, Netflix own method had a $rmse = 0.9514$. Consequently, the above decomposition constitutes a 0.92% improvement.

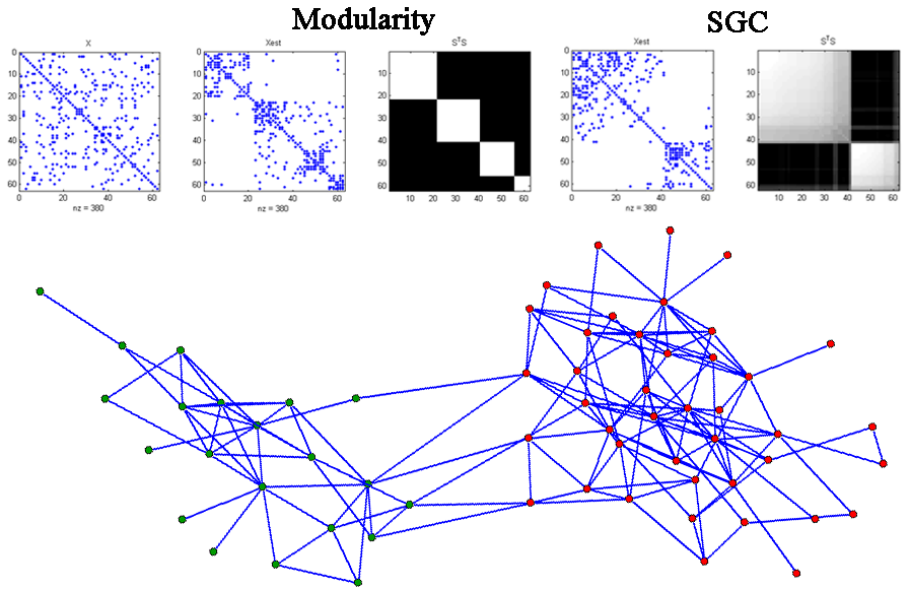


Figure 5.4: Result obtained by an SGC and modularity analysis of the dolphin network. **Top panel:** The raw network, the network permuted according to the clusters found as well as the density $\boldsymbol{S}^\top \boldsymbol{S}$ both for the modularity and SGC algorithm. Clearly, the SGC has identified 2 clusters and a few nodes belonging to both clusters, whereas the modularity has split the graph into 4 clusters despite a strong degree of overlap between the 3 last clusters found. **Bottom panel:** A plot of the dolphin network where vertices are marked according to the two clusters identified. The nodes overlapping are the same nodes set to belong to both clusters by the SGC algorithm. Since we had no prior information on the cluster densities ρ was set to the average density of the graph.

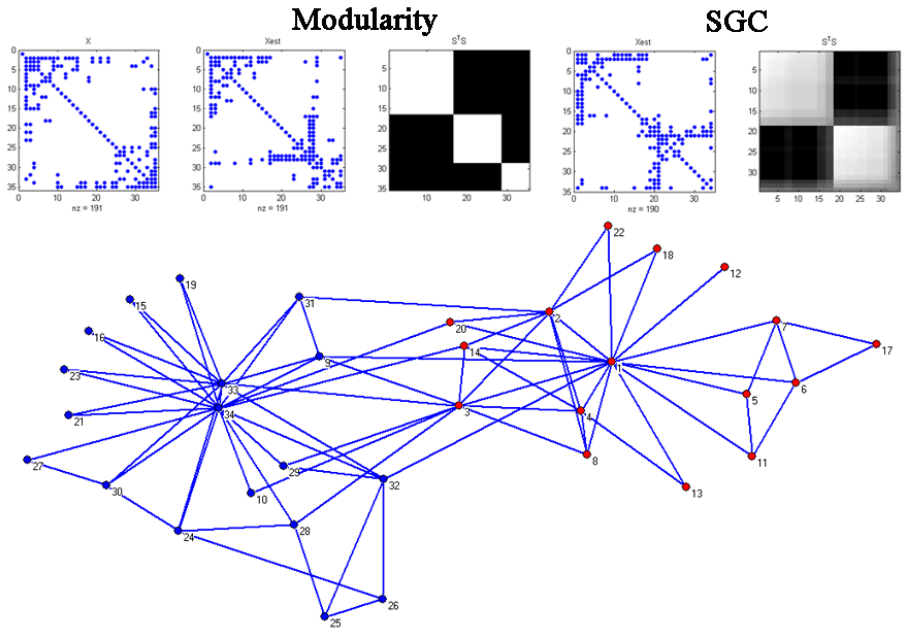


Figure 5.5: An analysis of Zachary's karate club. **Top panel:** The raw network, the network permuted according to the clusters found as well as the density $\boldsymbol{S}^\top \boldsymbol{S}$ both for the modularity and SQC algorithm. While the modularity identifies 3 clusters, two clusters with overlap are found by the SGC method. **Bottom panel:** The Network clustered according to the SGC algorithm. Indeed the correct two clusters have been found corresponding exactly to the nodes pertaining to the instructors (red nodes) and administrators (blue nodes). Since we had no prior information on the cluster densities ρ was set to the average density of the graph.

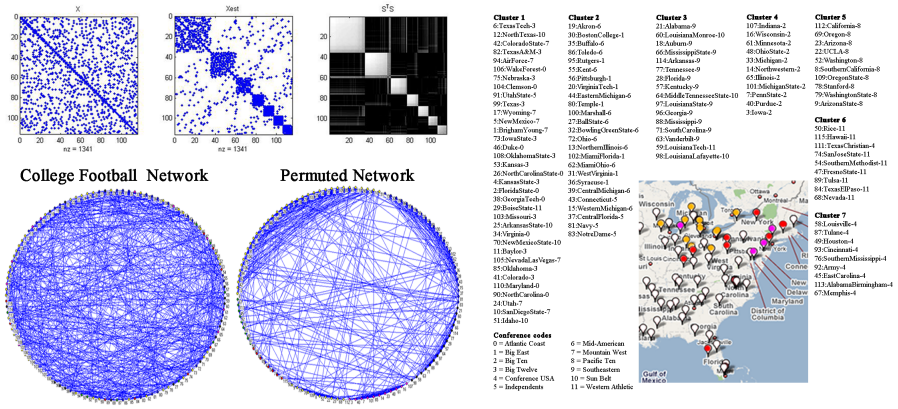


Figure 5.6: An SGC analysis of the college football network, ρ was set to the average density of the graph. **Top panel:** The clustering results. 7 clusters are identified and overlap between the clusters indicated in the $\boldsymbol{S}^\top \boldsymbol{S}$ matrix to the right. **Middle panel** A plot of the raw network as well as a display of the permuted result obtained by permuting the data according to $\boldsymbol{S}$. Clearly, the intra cluster connectivity is stronger than the inter cluster connectivity. To the bottom is given the nodes belonging to each cluster as well as their conference code. Clearly, the clusters 4, 5, 6, and 7 correspond to conference 2, 8, 11, 4 while cluster 3 has collapsed some of the teams from the sunbelt into southeastern. Notice how Louisiana Tech from the Western Athletic conference here has been clustered together with the remaining teams from Louisiana despite being in a different conference. Cluster 2 has collapsed conference 1, 5,and 6 marked in the map by red, orange and purple respectively hence forming a geographically close cluster. Cluster 1 has more or less collapsed the remaining conferences 0, 3, 7, and 10.

Chapter 6

Conclusion

"Success has many fathers, but failure is an orphan."
– Unknown

The guiding principle throughout this thesis has been simplicity in the decomposition. We have demonstrated several strategies to attain this goal. In factor analysis we saw that the classic methods of Quartimax and Varimax rotation was based on attaining sparse loadings of the factors by minimizing objectives related to kurtosis. This approach was strongly related to maximizing independence between factors as in ICA. Forming fully non-negative models as in Non-negative Matrix Factorization resulted in part based representations. However, part based representations are also strongly related to simplicity as the whole is modelled as the sum of its parts. In Sparse Coding the aim of simplicity was more directly formulated such that any decomposition was estimated as a tradeoff between reconstruction of the data and deviation from sparsity of the components. We saw these principles generalized to tensors in chapter 3 as well as to more flexible representations incorporating various types of invariance in chapter 4. In chapter 5 it was demonstrated how clustering data can be considered a decomposition forming the ultimate sparse representation. The k-means objective can be written as a matrix decomposition with l_0-norm constraint forming a highly sparse binary indicator matrix. For clustering of graphs an equivalent matrix decomposition could be formulated using the D_ρ metric. Thus, clustering problems can also be considered guided by sparse representations and as such strive for simplicity of the representation. As for Sparse

Coding the l_1-norm could be used as a convex proxy for the l_0-norm to attain these representations. Thus, parsimony, redundancy reduction, independence, part based representations and clustering can all be considered strategies for attaining the same goal (i.e., fathers of the same success). Namely, as stated by William of Ockham already in the 13th century that among several possible accounts of a phenomenon the simplest account is the best.

Perhaps the single most important algorithmic framework in this thesis was Sparse Coding. At the end of chapter 4 a highly overcomplete rotation and shift invariant analysis of natural images could be estimated using the principle of Sparse Coding. Here a dataset of size $512 \times 512 \times 10$ was attempted to be explained by a model with $(512+20-1)\times(512+20-1)\times 10\times 10\times 4+20\times 20\times 4$ variables (i.e., the model had more than 40 times as many free variables as observations). By imposing sparseness we demonstrated how this model could be solved such that features that strongly resemble the simple cell receptive fields of the brain was achieved. Thus, the principle of parsimony (i.e., sparseness of the solutions) is an important strategy when facing overcomplete representation, is strongly related to the information processing in the brain and can be considered an important strategy for the solution of inverse problems in general which has classically been solved using Tikhonov regularization [Hansen, 1992; Lawson and Hanson, 1974]. I.e., by imposing some form of l_2-norm regularization which does not admit sparse solutions. That sparse codes are an important principle for neural information processing is emphasized in [Graham and Field, 2006; Olshausen and Field, 2004]. We here quote the overwhelming amount of evidence given in [Graham and Field, 2006]

> *"...there are a number of studies suggesting that many neural systems utilize highly sparse codes. Deweese et al. [2003], recording from auditory neurons in the rat, have demonstrated that neurons in A1 can reliably produce a single spike in response to a sound. Evidence from olfactory systems in insects ([Perez-Orive et al., 2002; Theunissen, 2003]), somatosensory neurons in rat [Brecht and Sakmann, 2002], and recordings from rat hippocampus [Thompson and Best, 1989] all demonstrate highly sparse responses. Prefrontal cortex shows similar sparseness in behaving rhesus monkeys [Abeles, 1990]. As the authors of the latter study say, most areas of association cortex are "not carrying out any computations for the majority of the time....Motor neuron representations are often described as a population code, where it is proposed that the accuracy of a movement is guided by the degree of activity of a relatively large population of neurons (see [Georgopoulos et al., 1986]). Here too, we find evidence of sparse responses. Some motor neurons in layer 6 of rabbit motor cortex will produce just one spike during some movements [Beloozerova et al., 2003]. And stimulation of a single neuron in the rat is*

sufficient to deflect a whisker [Brecht and Sakmann, 2002]."

Thus, Sparse Coding is not only useful from a mathematical perspective but seems also to be an important principle for learning in biological systems in general.

While simplicity of the representation is attractive imposing too much sparseness generally results in highly localized features that do not generalize, see for instance Figure 4.12. Thus, while simplicity of the representation is attractive finding an optimal tradeoff between the simplicity of the representation and how well this representation generalizes is an open problem. In [Mørup et al., 2008e] we proposed to use an L-curve approach to investigate this tradeoff, see also Figure 4.8. Other approaches include evaluating information criteria such as AIC and BIC as well as approximating the evidence of the data as described in section 2.7. Somehow nature must have found a way to control the learning problem such that simplicity is evaluated relative to generalizability. As Graham and Field [2006] suggest

> *"...there is a penalty that applies to learning if the system is too sparse. An extremely sparse code (one in which neurons are highly selective for specific objects in specific poses, lighting etc.) would have neurons that fired quite rarely. In order to effectively learn about the world, any system must keep track of the relative probability of co-occurrences. No matter how a neural system keeps track of these co-occurrences, if they occur too rarely it would be impossible to determine whether any feature is statistically related to any other feature. We cannot learn about how "faces" behave in particular situations, if we have a neuron for every unique face. It is important the system be invariant at some level so that we can collapse across instances of the category. Most presentations of objects or events will occur just once or not at all during development if the object is defined too precisely....We therefore argue that although the evolution of large brains may allow a larger number of highly selective neurons, the constraints of learning require that the selectivity go hand in hand with a greater degree of invariance."*

Indeed improving the invariance of the representation improves component identifiability as demonstrated in chapter 4. However, even very flexible invariant representations can become too specific if there is not a built in control mechanism evaluating the generalizability. Thus, while sparseness, parsimony, reduction of redundancy etc. are important principles for Unsupervised Learning how to control what is the correct degree of sparseness/simplicity/reduction in redundancy is an important open problem. Without sound and efficient methods

to evaluate this important issue the "correct" internal representations can not be obtained and the trust and usefulness of the aforementioned decompositions becomes questionable. Thus, while efficient methods have been derived to decompose data into internal representations efficiently and correctly evaluating the right representation, i.e. the evidence of the model, is probably one of the most challenging problems facing researchers in Unsupervised Learning.

Yet another challenge for Unsupervised Learning is to formulate the "correct" models. Often the models considered are either too flexible or too restricted for the data at hand. In chapter 3 and 4 we considered a variety of extensions to the factor analysis based model. In general, understanding the processes underlying the generation of the data is crucial in formulating a good model that well extract the underlying hidden internal representation. Small extensions of models can significantly change the structure of the estimated components as demonstrated in Figure 4.5, 4.7 and 4.13. Thus, formulating the correct type of models with the adequate amount of flexibility to capture the underlying structure in the data is perhaps the most important aspect of Unsupervised Learning requiring a deep understanding of how the desired internal representation is formed, i.e. how the data is generated.

Appendix A

Appendix

A.1 Aspects of Least Squares Optimization in Factor Analysis

This appendix is dedicated to useful aspects of least squares optimization not described elsewhere in this thesis.

A.1.1 Cost function evaluation

The least squares cost function for the factor analysis model is given by

$$\begin{aligned} C_{LS} = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2 &= \frac{1}{2}\operatorname{trace}(\boldsymbol{X}\boldsymbol{X}^T) + \frac{1}{2}\operatorname{trace}(\boldsymbol{A}\boldsymbol{S}\boldsymbol{S}^\top\boldsymbol{A}^\top) - \operatorname{trace}(\boldsymbol{X}\boldsymbol{S}^\top\boldsymbol{A}^\top) \\ &= \frac{1}{2}\|\boldsymbol{X}\|_F^2 + \frac{1}{2}\mathbf{1}^\top(\boldsymbol{S}\boldsymbol{S}^\top) \bullet (\boldsymbol{A}^\top\boldsymbol{A})\mathbf{1} - \mathbf{1}^\top(\boldsymbol{X}\boldsymbol{S}^\top) \bullet \boldsymbol{A}\mathbf{1} \end{aligned}$$

Furthermore, the derivative of the least squares cost function with respect to $\boldsymbol{A}$ is given by

$$\nabla_{\boldsymbol{A}} C_{LS} = \boldsymbol{A}(\boldsymbol{S}\boldsymbol{S}^\top) - \boldsymbol{X}\boldsymbol{S}^\top \tag{A.1}$$

Equation this gradient to zero gives

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{S}^\top(\boldsymbol{S}\boldsymbol{S}^\top)^{-1} \tag{A.2}$$

Thus, to update $\boldsymbol{A}$ we need to calculate $\boldsymbol{X}\boldsymbol{S}^\top$ and $\boldsymbol{S}\boldsymbol{S}^\top$. As a result, all we need to compute to evaluate the cost function is $\frac{1}{2}\|\boldsymbol{X}\|_F^2$ and $\boldsymbol{A}^\top\boldsymbol{A}$. Hence, evaluating the Least Squares objective is inexpensive since most of the calculations required are obtained from the gradient and Hessian used to update the variables.

A.1.2 Weighted least squares and general l_p-norm optimization

The weighted Least Squares problem is given by

$$C_{WLS} = \sum_{i,j} \boldsymbol{Q}_{i,j}(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S})_{i,j}^2 \tag{A.3}$$

Such that $\boldsymbol{Q}$ weights the importance of the observations in $\boldsymbol{X}$. Differentiating C_{WLS} with respect to $\boldsymbol{A}$ and equating to zero we obtain

$$A_{i,:} = \boldsymbol{X}\,\mathrm{diag}(\boldsymbol{Q}_{i,:})\boldsymbol{S}^\top(\boldsymbol{S}\,\mathrm{diag}(\boldsymbol{Q}_{i,:})\boldsymbol{S}^\top)^{-1}. \tag{A.4}$$

We now observe that any l_p norm objective can be written in the form

$$\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_p^p = \sum_{i,j} |\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}|_{i,j}^p = \sum_{i,j} \frac{(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S})_{i,j}^2}{|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}|_{i,j}^{2-p}} \tag{A.5}$$

Thus, this can be solved as a weighted least squares problem where $\boldsymbol{Q}_{i,j} = |\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}|_{i,j}^{p-2}$.

A.1.3 Constrained Least Squares

A.1.3.1 Orthonormality

Solving the least squares problem $\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2$ with orthonormality constraint on $\boldsymbol{A}$, i.e. such that $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}$ can be done through the SVD of $\boldsymbol{X}\boldsymbol{S}^\top$. Let $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^\top] = \textsc{svd}(\boldsymbol{X}\boldsymbol{S}^\top)$ then $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^\top$ is a solution to the problem.

A.1.3.2 Unit l_p-norm Constraints

For sparse coding and other regularized least squares problem a trivial solution is to let the regularized variable go to zero while the remaining variables goes to infinity. To avoid this issue it is customary to enforce some form of norm

constraint on the un-regularized variables. I.e. to require that $\|\boldsymbol{A}\|_p = 1$ or $\|\mathbf{a}_d\|_p = 1$ that is that the norm of the matrix $\boldsymbol{A}$ or each component $\mathbf{a}_d$ have unit norm according to some l_p-norm (most often unit l_2-norm). In general there are two useful approaches to solve for this problem - either to use Lagrange multipliers to enforce the constraint or to reformulate the objective function such that it is invariant to the normalization.

Lagrange multiplier

Using Lagrange multipliers is a classic method in constrained optimization. Consider the following lagrange equation that enforces unit norm constraints on the columns of $\boldsymbol{A}$ by the lagrange multiplier λ_d

$$C_{\boldsymbol{A},\lambda} = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \sum_d \lambda_d(\mathbf{a}_d^\top \mathbf{a}_d - 1) \tag{A.6}$$

Differentiating the equation and equating to zero would yield the required equations. However, in [Lee et al., 2007] the following Lagrange dual method was proposed. i.e.

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{S}^\top(\boldsymbol{S}\boldsymbol{S}^\top + \mathrm{diag}(\boldsymbol{\lambda}))^{-1}. \tag{A.7}$$

Thus, $\boldsymbol{A}$ can be substituted to form an objective only in the variable $\boldsymbol{\lambda}$ as

$$C_\lambda = trace(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}\boldsymbol{S}^\top(\boldsymbol{S}\boldsymbol{S}^\top - \mathrm{diag}(\boldsymbol{\lambda}))^{-1}\boldsymbol{S}\boldsymbol{X}^\top) - \sum_d \lambda_d \tag{A.8}$$

The gradient and Hessian can be calculated for this objective and $\boldsymbol{\lambda}$ estimated using for instance the Newton-Raphson method. Unfortunately, there is a risk of ending in a suboptimal solution for λ since this is a non-convex objective - thus, there is no guarantee that the correct $\boldsymbol{\lambda}$ are found every time. Similarly, normalizing by the norm of the full matrix $\boldsymbol{A}$ gives

$$C_{\boldsymbol{A},\lambda} = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2 + \lambda(\sum_d \mathbf{a}_d^\top \mathbf{a}_d) - 1 \tag{A.9}$$

Again, $\boldsymbol{A}$ can be substituted to form an objective only in the variable λ as

$$C_\lambda = trace(\boldsymbol{X}\boldsymbol{X}^\top - \boldsymbol{X}\boldsymbol{S}^\top(\boldsymbol{S}\boldsymbol{S}^\top - \lambda\boldsymbol{I})^{-1}\boldsymbol{S}\boldsymbol{X}^\top) - \lambda. \tag{A.10}$$

Normalization Invariance

In the normalization invariance approach proposed [Eggert and Körner, 2004] the objective function is reformulated such that it is invariant to a given normalization. Define the l_p-normalization invariant variable $\widehat{\boldsymbol{a}}_{\boldsymbol{d}} = \frac{\boldsymbol{a_d}}{\|\boldsymbol{a_d}\|_p}$ for invariance to column-wise normalization and $\widehat{\boldsymbol{A}} = \frac{\boldsymbol{A}}{\|\boldsymbol{A}\|_p}$ for invariance to normalization by the full matrix. Then the objective function is rewritten as

$$C_{\boldsymbol{A}} = \frac{1}{2}\|\boldsymbol{X} - \widehat{\boldsymbol{A}}\boldsymbol{S}\|_F^2 \tag{A.11}$$

Hence, the objective is invariant to the normalization of $\boldsymbol{A}$. By differentiating this objective the gradient and Hessian can be calculated and $\mathbf{A}$ updated for instance by the Newton-Raphson procedure. Notice, this objective is no longer necessarily convex for fixed values of $\boldsymbol{S}$.

A.1.3.3 Simplicial Optimization

Consider the problem $\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2$ subject to $\sum_d \boldsymbol{S}_{d,j} = 1$, $\boldsymbol{S}_{d,j} \geq 0$ that is the solution of each column of $\boldsymbol{S}$ resides on the unit simplex. This problem can be separated into the following J subproblems.

$$\begin{array}{ll} \text{minimize} & \|\mathbf{x}_j - \boldsymbol{A}\mathbf{s}_j\|_F^2 \\ \text{subject to} & \|\mathbf{s}_j\|_1 = 1,\ \mathbf{s}_j \geq 0 \end{array} \tag{A.12}$$

Defining the matrix $\boldsymbol{Q} = \boldsymbol{A} - \mathbf{x}_j\mathbf{1}^\top$ the problem above is equivalent to

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{s}_j^\top \boldsymbol{Q}^\top \boldsymbol{Q}\mathbf{s}_j \\ \text{subject to} & \|\mathbf{s}_j\|_1 = 1,\ \mathbf{s}_j \geq 0 \end{array} \tag{A.13}$$

This cost is trivially minimized when $\mathbf{s}_j = \mathbf{0}$ thus to make sure $\|\mathbf{s}_j\|_1 = 1$ we add the lagrange multiplier $-\lambda\|\mathbf{s}_j\|_1$ to the objective giving

$$\underset{\mathbf{s}_j}{\operatorname{argmin}} \frac{1}{2}\mathbf{s}_j^\top \boldsymbol{Q}^\top \boldsymbol{Q}\mathbf{s}_j - \lambda\mathbf{1}^\top\mathbf{s}_j. \tag{A.14}$$

This can be solved using the NNQP. The solution is given as $\widehat{\mathbf{s}}_{A,j} = \lambda(\boldsymbol{Q}\boldsymbol{Q}^\top)^{-1}_{A,A}\mathbf{1}_A$, where A index over active elements. Thus, it can be seen that λ simply scales the solution such that the solution is given as $\mathbf{s}_j = \frac{\widehat{\mathbf{s}}_j}{\|\widehat{\mathbf{s}}_j\|_1}$.

A.1.3.4 Smoothness Constraints

Traditionally, smoothness has been imposed penalizing changes over time in the signal based on penalty terms such as local estimate of the temporal derivative, (i.e., $\|s_t - s_{t-1}\|_F^2$) or curvature, (i.e., $\|s_{t-1} + s_{t+1} - 2s_t\|_F^2$) [Bro, 1998; Hastie and Tibshirani, 1990]. This can in the frequency domain be expressed as

$$\|\widetilde{w}_f\mathbf{s}_f\|_F^2 \tag{A.15}$$

where $\widetilde{w}_f = 1 - e^{2\pi f/N}$ and $\widetilde{w}_f = e^{2\pi f/N} + e^{-2\pi f/N} - 2$ respectively. Smoothness constraints in the time domain corresponds to reduced high frequency content. Hence smoothness can be imposed by penalizing high frequency regions of the

components. The above smoothness penalty has a closed form solution for least squares optimization in the frequency domain. I.e.

$$C_{SmoothLS}(\widetilde{\boldsymbol{S}}) = \frac{1}{2}\sum_f \|\widetilde{\mathbf{x}}_f - \boldsymbol{A}\widetilde{\mathbf{s}}_f\|_F^2 + \|\widetilde{w}_f\widetilde{\mathbf{s}}_f\|_F^2 \tag{A.16}$$

Has the solution

$$\widetilde{\mathbf{s}}_f = (\boldsymbol{A}^\top\boldsymbol{A} + \widetilde{w}_f\widetilde{w}_f^*\boldsymbol{I})^{-1}\boldsymbol{A}^\top\widetilde{\mathbf{x}}_f \tag{A.17}$$

Smoothness can be combined with constraints in the time domain through the TFGM, see also [Madsen et al., 2008]. Notice, smoothness as imposed by the Frobenius norm does not guarantee uniqueness of the decomposition since the alternative solution given by the orthonormal matrix $\boldsymbol{Q}$, i.e. $\widetilde{\boldsymbol{S}}' = \boldsymbol{Q}\widetilde{\boldsymbol{S}}$ and $\widehat{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{Q}^\top$ gives identical value of $C_{SmoothLS}$, i.e., $\|\widetilde{w}_f\widetilde{\mathbf{s}}_f\|_F^2 = \widetilde{w}_f\widetilde{w}_f^*\widetilde{\mathbf{s}}_f^\top\widetilde{\mathbf{s}}_f = \widetilde{w}_f\widetilde{w}_f^*\widetilde{\mathbf{s}}_f^\top\boldsymbol{Q}^\top\boldsymbol{Q}\widetilde{\mathbf{s}}_f = \|\widetilde{w}_f\widetilde{\mathbf{s}}_f'\|_F^2$.

A.1.3.5 Missing Values

An important problem in least squares estimation is the handling of missing data. Traditionally, two strategies have been employed: Imputation and Marginalization.

Imputation

In Imputation the missing values are estimated in each iteration (i.e., imputed) based on the maximum likelihood estimate of the model such that the missing values do not contribute to the objective function. I.e.

$$\boldsymbol{X}_{i,j}^{missing} = \sum_d \boldsymbol{A}_{i,d}\boldsymbol{S}_{d,j}. \tag{A.18}$$

$\boldsymbol{A}$ and $\boldsymbol{S}$ are then solved for by regular analysis including all entries in $\boldsymbol{X}$ where missing entries are replaced by their estimated values.

Marginalization

By marginalizing the missing values in the objective function the missing values do not contribute in the estimation of $\boldsymbol{A}$ and $\boldsymbol{S}$. Thus, by considering the weighted least squares objective

$$C_{WLS} = \sum_{i,j}\boldsymbol{Q}_{i,j}(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S})_{i,j}^2, \tag{A.19}$$

values that are missing in $\boldsymbol{X}$ are given weight zero in the corresponding indices of $\boldsymbol{Q}$ whereas values present are given weight one in $\boldsymbol{Q}$. However, other values in $\boldsymbol{Q}$ are conceivable. For instance, in text data mining let $\boldsymbol{X}^{term\times document}$ be the term by document occurrence matrix. It is here not always clear whether entries that are zero are a result of the term not being present in the document

or because the document was too short for the term to appear. Thus, zero entries can either be considered missing or having a value of zero. By setting the corresponding values in $\boldsymbol{Q}$ between zero and one it is possible to interpolate between these two interpretations of zero entries in the term-document matrix. In fact, the weighted least squares can be interpreted in terms of a probability distribution where each data point is drawn with varying certainty given by the variance $Q_{i,j}^{-1}$.

A.2 Objective Functions and Likelihood Functions

According to Bayes' Theorem we have,

$$P(\boldsymbol{\theta}|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{X})} \tag{A.20}$$

Where $P(\boldsymbol{X}) = \int_{\boldsymbol{\theta}} P(\boldsymbol{X}|\boldsymbol{\theta})P(\boldsymbol{\theta})$. There are several approaches to estimating parameters in the Bayesian setting [Petersen, 2005]. We will presently consider Maximum Likelihood and Maximum a Posteriori estimation

- **Maximum Likelihood (ML):** $\frac{\partial \log[P(\boldsymbol{X}|\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}} = \mathbf{0}$ that is the parameters in $\boldsymbol{\theta}$ are found so as to maximize the probability of the observed data set.
- **Maximum a Posteriori (MAP):** $\frac{\partial \log[P(\boldsymbol{\theta}|\boldsymbol{X})]}{\partial \boldsymbol{\theta}} = \mathbf{0}$ that is the parameters in $\boldsymbol{\theta}$ is found based on the assumed prior distributions of $\boldsymbol{\theta}$ enabling to form the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{X})$ using Bayes' Theorem. If the prior is uninformative this estimation becomes identical to ML.

For the model $\boldsymbol{X} \approx \boldsymbol{AS}$ we have using Bayes' theorem

$$P(\boldsymbol{A}, \boldsymbol{S}|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{S})P(\boldsymbol{A})P(\boldsymbol{S})}{P(\boldsymbol{X})} \tag{A.21}$$

Where $P(\boldsymbol{X}) = \int_{\boldsymbol{A},\boldsymbol{S}} P(\boldsymbol{X}|\boldsymbol{A}, \boldsymbol{S})P(\boldsymbol{A})P(\boldsymbol{S})$ is a normalization constant also denoted the marginal likelihood or evidence. While the evidence is important for model evaluation it is an intractable quantity to compute for almost all models of interest [Beal, 2003]. Traditionally, the marginal likelihood has been approximated either using analytical methods, for example the Laplace approximation, variational Bayes also named ensemble learning or via sampling-based approaches such as Markov Chain Monte Carlo[Beal, 2003].

Assuming further that each observation and source (i.e., $\mathbf{x}_j$ and $\mathbf{s}_j$) are i.i.d. we get

$$P(\boldsymbol{A}, \boldsymbol{S}|\boldsymbol{X}) = \prod_j \frac{P(\mathbf{x}_j|\boldsymbol{A}, \mathbf{s}_j)P(\boldsymbol{A})P(\mathbf{s}_j)}{P(\boldsymbol{x}_j)} \tag{A.22}$$

again, $P(\boldsymbol{x}_j) = \int_{\boldsymbol{A},\mathbf{s}_j} P(\boldsymbol{X}|\boldsymbol{A},\mathbf{s}_j)P(\boldsymbol{A})P(\mathbf{s}_j)$ is a normalization constant. The log-likelihood is given by

$$\log[P(\boldsymbol{A},\boldsymbol{S}|X)] \propto \sum_j \log[P(\mathbf{x}_j|\boldsymbol{A},\mathbf{s}_j)] + \log[P(\boldsymbol{A})] + \log[P(\mathbf{s}_j)]. \tag{A.23}$$

Hence, assuming Gaussian noise with covariance $\sigma^2\boldsymbol{I}$ we find

$$P(\mathbf{x}_j|\boldsymbol{A},\mathbf{s}_j) = \frac{1}{2\pi|\sigma\boldsymbol{I}|} e^{-\frac{1}{2\sigma^2}(x_j-\boldsymbol{A}\mathbf{s}_j)^\top \boldsymbol{I}^{-1}(x_j-\boldsymbol{A}\mathbf{s}_j)} \tag{A.24}$$

which yield the regular least squares objective. Thus, regularization terms on $\boldsymbol{A}$ and $\boldsymbol{S}$ correspond to taking the log of given prior distributions of $\boldsymbol{A}$ and $\boldsymbol{S}$, i.e., $\log[P(\boldsymbol{A})]$ and $\log[P(\mathbf{s}_j)]$ in equation (A.23). Thus, the objective functions of this thesis including regularization can all be stated in terms of Bayesian learning using MAP estimation while regular least squares estimation corresponds to ML estimation based on a Gaussian noise model.

Index

Bibliography

Abeles, M., V. E. . B. H. (1990). Firing patterns of single units in the prefrontal cortex and neural network models. *Network: Computation in Neural Systems*, 1(1):13–25.

Acar, E., Aykut-Bingol, C., Bingol, H., Bro, R., and Yener, B. (2007). Multiway analysis of epilepsy tensors. *Bioinformatics*, 23:i10–18.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.

Albright, T. D. (1984). Direction and orientation selectivity of neurons in visual area mt of the macaque. *Journal of Neurophysiology*, 52(6):1106–1130.

Amari, S.-I. and A., C. (1998). Adaptive blind signal processing - neural network approaches. *Proceedings of the IEEE*, 86(10):2026–2048.

Andersen, A. H. and Rayens, W. S. (2004). Structure-seeking multilinear methods for the analysis of fmri data. *NeuroImage*, 22:728–739.

Andersson, C. A. and Bro, R. (1998). Improving the speed of multi-way algorithms: Part i. tucker3. *Chemometrics and Intelligent Laboratory Systems*, 42:93–103.

Anemuller, J., Sejnowski, T. J., and Makeig, S. (2003). Complex independent component analysis of frequency-domain electroencephalographic data. *Neural Networks*, 16(9):1311–1323.

Arnfred, S., Hansen, L., Parnas, J., and Mørup, M. (2007). Proprioceptive evoked gamma oscillations. *Brain Research*, 1147:167–174.

Arnfred, S., Hansen, L., Parnas, J., and Mørup, M. (2008). Regularity increases middle-latency evoked and late induced beta brain response following proprioceptive stimulation. *accepted for publication, Brain Research*.

Attias, H. and Schreiner, C. (1998). Blind source separation and deconvolution: the dynamic component analysis algorithm. *Neural Computation*, 10(6):1373–1424.

Bach, F. R. and Jordan, M. I. (2004). Learning spectral clustering. In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Banerjee, A., Merugu, S., Dhillon, I. S., and Ghosh, J. (2005). Clustering with bregman divergences. *J. Mach. Learn. Res.*, 6:1705–1749.

Barlow, H. (1961). Possible principles underlying the transformations of sensory messages. *Sensory Communication, MIT Press*, pages 217–234.

Barlow, H. (1996). The role of redundancy. *Perception as Bayesian Inference By David C. Knill, Whitman Richard*, page 213.

Beal, M. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of London.

Beckmann, C. and Smith, S. (2005). Tensorial extensions of independent component analysis for multisubject fmri analysis. *NeuroImage 25*, pages 294–311.

Bell, A. J. and Sejnowski, T. J. (1995). An information maximization approach to blind source separation and blind deconvolution. *Neural Computation*, 7:1129–1159.

Beloozerova, I., Sirota, M., and Swadlow, H. (2003). Activity of different classes of neurons of the motor cortex during locomotion. *J. Neurosci.*, 23:1087–1097.

Berkhin, P. (2002). Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA.

Berman, A. and Plemmons, R. (1979). Nonnegative matrices in the social sciences. *Academic, New York*.

Blumensath, T. and Davies, M. (2004a). On shift-invariant sparse coding. *International Conference on Independent Component Analysis and Blind Source Separation*, 26:1205–1212.

Blumensath, T. and Davies, M. (2004b). On shift-invariant sparse coding. *International Conference on Independent Component Analysis and Blind Source Separation*, 26:1205–1212.

Bödvarsson, B., Hansen, L. K., Svarer, C., and Knudsen, G. M. (2007). NMF on positron emission tomography. In *International conference on acoustics, speech and signal processing 2007, ICASSP.*

Brecht, M. and Sakmann, B. (2002). Dynamic representation of whisker deflection by synaptic potentials in spiny stellate and pyramidal cells in the barrels and septa of layer 4 rat somatosensory cortex. *J. Physiol. (Lond.)*, 543:49–70.

Bro, R. (1998). *Multi-way Analysis in the Food Industry.* PhD thesis.

Bro, R., Andersson, C., and Kiers, H. (1999). Parafac2 - part ii. modeling chromatographic data with retention time shifts. *Journal of Chemometrics*, 13:295–309.

Bro, R. and Andersson, C. A. (2000). The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52:1–4.

Bro, R. and de Jong, S. (1997). A fast non-negativity-constrained least squares algorithm. *J. of Chemometrics*, 11(5):393–401.

Bro, R. and Kiers, H. A. L. (2003). A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics*, 17(5):274–286.

Buxton, R. B., Wong, E. C., and Frank, L. R. (1998). Dynamics of blood flow and oxygenation changes during brain activation: the balloon model. *Magn Reson Med*, 39(6):855–864.

Cardoso, J. (1997). Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114.

Cardoso, J.-F. (1998). Blind signal separation: statistical principles. *Proceedings of the IEEE*, 9(10):2009–2025.

Carroll, J. B. (1953). An analytical solution for approximating simple structure in factor analysis. *Psychometrika*, 18:23–38.

Carroll, J. D. and Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319.

Cattell, R. (1944). Parallel proportional profiles and other principles for determining the choice of factors by rotation. *Psychometrika*, 9:267–283.

Cattell, R. (1952). The three basic factor-analytic research designs – their interrelations and derivatives. *Psychological Bulletin*, 49:499–520.

Cichocki, A., Zdunek, R., and Amari, S. (2006). Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. *6th International Conference on Independent Component Analysis and Blind Signal Separation*, pages 32–39.

Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., and Amari, S.-i. (2007). Nonnegative tensor factorization using alpha and beta divergencies. *ICASSP*.

Cohen, J. E. and Rothblum, U. G. (1993). Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *LINEAR ALGEBRA AND ITS APPLICATIONS*, 190:149–168.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, 36:287–314.

Crawford, C. B. and Ferguson, G. A. (1970). A general rotation criterion and its use in orthogonal rotation. *Psychometrika*, 35:321–332.

Dantzig, G. B. (1963). *Linear programming*. Princeton University Press.

De Vos, M., Vergult, A., De Lathauwer, L., De Clercq, W., Van Huffel, S., Dupont, P., Palmini, A., and Van Paesschen, W. (2007). Canonical decomposition of ictal scalp eeg reliably detects the seizure onset zone. *NeuroImage*, 37(3):844–54.

De Vos M., De Lathauwer L., V. H. S. (2007). Imposing independence constraints in the cp model. *Proc. of the 7th international conference on independent component analysis (ICA'07)*.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Deweese, M. R., Wehr, M., and Zador, A. M. (2003). Binary spiking in auditory cortex. *J Neurosci*, 23(21):7940–7949.

Dhillon, I. and Sra, S. (2006). Generalized nonnegative matrix approximations with bregman divergences. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 283–290. MIT Press, Cambridge, MA.

Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. *Proc. SIAM Internat. Conf. Data Min. (SDM'05)*, pages 606–610.

Ding, C., Li, T., and Jordan, M. I. (2006a). Convex and semi-nonnegative matrix factorizations. *LBNL Tech Report 60428*.

Ding, C., Li, T., Peng, W., and Park, H. (2006b). Orthogonal nonnegative matrix t-factorizations for clustering. In *Internat. Conf. on Knowledge Discov. and Data Min. (KDD'06)*, pages 126–135, New York, NY, USA. ACM.

Donchin, E. and Heffley, E. (1978). Multivariate analysis of event-related potential data: A tutorial review. *Multidisciplinary perspectives in event-related brain potential research*, pages pp. 555–572.

Donoho, D. (2006). For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829.

Donoho, D. and Stodden, V. (2004). When does nonnegative matrix factorization give a correct decomposition into parts? In Thrun, S., Saul, L., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA.

Du, S., Sajda, P., Stoyanova, R., and Brown, T. R. (2005). Recovery of metabolomic spectral sources using non-negative matrix factorization. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology*, pages 4731–4734.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499.

Eggert, J. and Körner, E. (2004). Sparse coding and nmf. In *Neural Networks*, volume 4, pages 2529–2533.

Eggert, J., Wersing, H., and Körner, E. (2004). Transformation-invariant representation and nmf. In *Neural Networks*, volume 4, pages 2535– 2539.

Emile, B. and Comon, P. (1998). Estimation of time delays between unknown colored signals. *Signal Processing*, 68(1):93–100.

Ferguson, G. (1954). The concept of parsimony in factor analysis. *Psychometrika*, 19:281–290.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305.

Field, A. S. and Graupe, D. (1991). Topographic component (parallel factor) analysis of multichannel evoked potentials: Practical issues in trilinear spatiotemporal decomposition. *Brain Topography*, 3(4):407–423.

Field, D. (1994). What is the goal of sensory coding? *Neural Computation*, 6:559–601.

FitzGerald, D. and Coyle, E. (2006). Sound source separation using shifted non-negative tensor factorisation. In *ICASSP2006*.

Georgopoulos, A., Schwartz, A., and Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, 233:1416–1419.

Ghahramani, Z. and Roweis, S. (1999). Probabilistic models for unsupervised learning. Tutorial presented at the NIPS Conference.

Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826.

Gobinet, C., Perrin, E., and Huez, R. (2004). Application of non-negative matrix factorization to fluorescence spectroscopy. *EUSIPCO*, pages 1095–1098.

Golub, G. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computation*. Johns Hopkins Studies in Mathematical Sciences, 3 edition.

Graham, D. J. and Field, D. J. (2006). Sparse coding in the neocortex. *Talk at Department of Psychology, Cornell University, Ithaca, NY 14817*.

Gregory, D. A. and Pullman, N. J. (1983). Semiring rank: Boolean rank and nonnegative rank factorization. *J. Combin. Inform. System Sci.*, 3:223–233.

Griskova, I., Mørup, M., Parnas, J., Ruksenas, O., and Arnfred, S. (2007). The amplitude and phase precision of 40 hz auditory steady-state response depend on the level of arousal. *Experimental Brain Research*, 183(1):133–138.

Gurden, S. P., Westerhuis, J. A., Bijlsma, S., and Smilde, A. K. (2001). Modelling of spectroscopic batch process data using grey models to incorporate external information. *Journal of Chemometrics*, 15:101–121.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 127(2):301–354.

Hansen, L. and Larsen, J. (1996). Unsupervised learning and generalization.

Hansen, L., Larsen, J., Nielsen, F., Strother, S., Rostrup, E., Savoy, R., Lange, N., Sidtis, J., Svarer, C., and Paulson, O. (1999). Generalizable patterns in neuroimaging: how many principal components? *Neuroimage*, 9:534–544.

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4):561–580.

Harshman, R. (1978). Models for analysis of asymmetrical relationships among n objects or stimuli. *First Joint Meeting of the Psychometric Society and the Society for Mathematical Psychology, McMaster University, Hamilton, Ontario*.

Harshman, R., Hong, S., and Lundy, M. (2003a). Shifted factor analysis part i: Models and properties. *Journal of Chemometrics*, 17:363–378.

Harshman, R., Hong, S., and Lundy, M. (2003b). Shifted factor analysis part ii: Algorithms. *Journal of Chemometrics*, 17:379–388.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84.

Harshman, R. A. (1972a). Determination and proof of minimum uniqueness conditions for parafac1. *UCLA Working Papers in Phonetics*, (22):111–117.

Harshman, R. A. (1972b). Parafac2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22:30–47.

Harshman, R. A. and Lundy, M. E. (1984). Data preprocessing and the extended parafac model. *In: Law, H. G., Snyder, Jr., C. W., Hattie, J. A., and McDonald, R. P. (eds.), Research Methods for Multimode Data Analysis, Praeger, New York,*, pages 216–281.

Harshman, R. A. and Lundy, M. E. (1996). Uniqueness proof for a family of models sharing features of tucker's three-mode factor analysis and parafac/candecomp. *Psychometrika*, 61:133–154.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Hastie, T. and Tibshirani, R. (1990). Generalized additive models.

Hateren, J. and Van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc.R.Soc.Lond. B*, 265:359–366.

Have, A. S., Girolami, M. A., and Larsen, J. (2006). Clustering via kernel decomposition. *IEEE Transactions on Neural Networks*, 17(1):256–264.

Hinton, G. and Sejnowski, T. J., editors (1999). *Unsupervised Learning Foundations of Neural Computation*. MIT Press.

Hitchcock, F. L. (1927a). The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys. Camb.*, pages 164–189.

Hitchcock, F. L. (1927b). Multiple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys. Camb.*, pages 39–70.

Højen-Sørensen, P. A., Winther, O., and Hansen, L. K. (2002). Mean field approaches to independent component analysis. *Neural Computation*, 14:889–918.

Hong, S. and Harshman, R. A. (2003). Shifted factor analysis part iii: N-way generalization and application. *Journal of Chemometrics*, 17:389–399.

Hoyer, P. (2002). Non-negative sparse coding. *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*.

Hubel, D. (1995). *Eye, Brain and Vision*. http://hubel.med.harvard.edu/.

Hyvärinen, A. and Hoyer, P. (2000). Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12:1705–1720.

Hyvarinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley and Sons.

Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128.

Jensen, A. R. (1994). Spearman, charles edward. *R. J. Sternberg (Ed.), Encyclopedia of intelligence*, 1:1007–1014.

Jensen, T. and Johansen, B. (19xx). Tåken letter (the fog is lifting) for flute and harp composed by carl nielsen. Naxos.

Jia, K. and Gong, S. (2005). Multi-modal tensor face for simultaneous super-resolution and recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1683–1690, Washington, DC, USA. IEEE Computer Society.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrica*, 23:187–200.

Kemp, C., Griffiths, T. L., and Tenenbaum, J. B. (2004). Discovering latent classes in relational data. Technical report.

Kiers, H. A. (1998). Joint orthomax rotation of the core and component matrices resulting from three-mode principal component analysis. *Journal of Classification*, 15:245–263.

Kim, K. M., Watabe, H., Shidahara, M., Ahn, J. Y., Choi, S., Kudomi, N., Hayashida, K., Miyake, Y., and Iida, H. (2001). Noninvasive estimation of cerebral blood flow using image-derived carotid input function in $h_2^{15}o$ dynamic pet. volume 3, pages 1282–1285.

Kolda, T. and Bader, B. (2008). Tensor decompositions and applications. *SIAM Review, to appear*.

Krijnen, W. P. and Berge, J. M. F. t. (1992). A constrained parafac method for positive manifold data. *Applied Psychological Measurement*, 16:295–305.

Kruskal, J. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18:95–138.

Kruskal, J. B. (1976). More factors than subjects, tests and treatments: An indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3).

Lathauwer, L. D., Moor, B. D., and Vandewalle, J. (2000). Multilinear singular value decomposition. *SIAM J. MATRIX ANAL. APPL.*, 21(4):1253–1278.

Laurberg, H., Christensen, M. G., Plumbley, M. D., Hansen, L. K., and Jensen, S. H. (2007). Theorems on positive data: On the uniqueness of nmf. *submitted, Computational Intelligence and Neuroscience*.

Lawson, C. and Hanson, R. (1974). *Solving Least Squares Problems*, volume 15 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1995.

Lee, D. and Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91.

Lee, D. D. and Seung, H. S. (2000). Algorithms for nonnegative matrix factorization. In *NIPS*, pages 556–562.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2007). Efficient sparse coding algorithms. *In Proceedings of the Neural Information Processing Systems (NIPS)*, 19.

Lee, J. S. L., Lee, D. D., Choi, S., Park, K. S., and Lee, D. S. (2001). Non-negative matrix factorization of dynamic images in nuclear medicine. *IEEE Nuclear Science Symposium and Medical Imaging Conference*, pages 2027–2030.

Lee, T.-W., Girolami, M., Bell, A., and Sejnowski, T. (2000). A unifying information-theoretic framework for independent component analysis. *International Journal on Mathematical and Computer Models*, 39:1–21.

Lehmann, S. and Hansen, L. K. (2007). Deterministic modularity optimization. *The European Physical Journal B*, 60(1):83–88.

Lewicki, M. S. and Sejnowski, T. J. (1999). Coding time-varying signals using sparse shift-invariant representations. *Adv. Neural Inform. Process. Systems (NIPS'99)*, 11:730–736.

Lim, L.-H. and Golub, G. (2006). Nonnegative decomposition and approximation of nonnegative matrices and tensors. *SCCM Technical Report, 06-01, forthcoming, 2006.*

Lin, C.-J. (2007). Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., and Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. can geographic isolation explain this unique trait? *Behav. Ecol. and Sociobiology*, 54(4):396–405.

M. Hofman, J. and Wiggins, C. H. (2007). A bayesian approach to network modularity. *Technical Report.*

Madsen, K., Hansen, L., and Mørup, M. (2008). Combining time and frequency domain optimization: Shifts, convolution and smoothness in factor analysis type decompositions. *submitted, Journal of Signal Processing Systems special issue on MLSP2007.*

Mahoney, M. W., Maggioni, M., and Drineas, P. (2006). Tensor-cur decompositions for tensor-based data. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 327–336, New York, NY, USA. ACM.

Makeig, S., Bell, A. J., Jung, T.-P., and Sejnowski, T. J. (1996). Independent component analysis of electroencephalographic data. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 145–151. The MIT Press.

Makeig, S., Jung, T.-P. ´and Bell, A., Ghahremani, D., and Sejnowski, T. (1997). Blind separation of auditory event-related brain responses into independent components. *Proc. Natl. Acad. Sci. USA*, 94:10979–10984.

Makeig, S., Westerfield, M., Jung, T., Covington, J., Townsend, J., Sejnowski, T., and Courchesne, E. (1999). Functionally independent components of the late positive event-related potential during visual spatial attention. *J. Neurosci.*, 19:2665–2680.

Makeig, S., Westerfield, M., Jung, T.-P., Enghoff, S., Townsend, J., Courchesne, E., and Sejnowski, T. J. (2002). Dynamic brain sources of visual evoked responses. *Science*, 295(5555):690–694.

McKeown, M. J., Hansen, L. K., and Sejnowski, T. J. (2003). Independent component analysis of functional mri: what is signal and what is noise? *Current Opinion in Neurobiology*, 13(5):620–629.

McKeown, M. J., Jung, T. P., Makeig, S., Brown, G., Kindermann, S. S., Lee, T. W., and Sejnowski, T. J. (1998). Spatially independent activity patterns in functional MRI data during the stroop color-naming task. *Proc Natl Acad Sci U S A*, 95(3):803–810.

Miwakeichi, F., Martínez-Montes, E., Valdés-Sosa, P. A., Nishiyama, N., Mizuharam, H., and Yamaguchi, Y. (2004). Decomposing eeg data into space time frequency components using parallel factor analysis. *NeuroImage*, 22(3):1035–1045.

Möcks, J. (1988). Topographic components model for event-related potentials and some biophysical considerations. *IEEE Trans. Biomed. Eng.*, 35:482–484.

Molgedey, L. and Schuster, H. (1994). Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(34):3634–3637.

Moon, T. (1996). The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60.

Mørup, M. and Clemmensen, L. H. (2007). Multiplicative updates for the lasso,. *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 33–38.

Mørup, M., Hansen, L., Arnfred, S., Lim, L.-K., and Madsen, K. (2008a). Shift invariant multilinear decomposition of neuroimaging data. *submitted NeuroImage*.

Mørup, M., Hansen, L., and Arnfred, S. M. (2007a). Erpwavelab a toolbox for multi-channel analysis of time-frequency transformed event related potentials. *Journal of Neuroscience Methods*, 161(361-368).

Mørup, M., Hansen, L., and Arnfred, S. M. (2008b). Algorithms for sparse nonnegative tucker, accepted for publication. *to appear, Neural Computation*.

Mørup, M., Hansen, L., and Lim, L.-H. (2008c). Simplicial clustering. *Submitted International Conference on Machine Learning (ICML2008)*.

Mørup, M., Hansen, L. K., Hermann, C. S., Parnas, J., and Arnfred, S. M. (2006a). Parallel factor analysis as an exploratory tool for wavelet transformed event-related eeg. *NeuroImage*, 29(3):938–947.

Mørup, M., Hansen, L. K., Parnas, J., and Arnfred, S. M. (2006b). Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization. Technical report.

Mørup, M., Madsen, K., and Hansen, L. (2008d). Approximate l0 constrained non-negative matrix and tensor factorization. *accepted ISCAS 2008 special session on Non-negative Matrix and Tensor Factorization and Related Problems.*

Mørup, M., Madsen, K. H., and Hansen, L. K. (2007b). Shifted independent component analysis. *Independent Component Analysis and Blind Signal Separation*, pages 89–96.

Mørup, M., Madsen, K. H., and Hansen, L. K. (2007c). Shifted non-negative matrix factorization. *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 139–144.

Mørup, M. and Schmidt, M. N. (2008). Efficient rotation and shift invariant sparse coding. *Submitted International Conference on Machine Learning (ICML2008).*

Mørup, M., Schmidt, M. N., and Hansen, L. K. (2008e). Shift invariant sparse coding of image and music data. *submitted, Journal of Machine Learning Research (JMLR), 2007.*

Murakami, T. and Kroonenberg, P. M. (2003). Three-mode models and individual differences in semantic differential data. *Multivariate Behavioral Research*, 38(2):247–283.

Neuhaus, J. . and Wrigley, C. (1954). The quartimax method: an analytical approach to orthogonal simple structure. *Brit. J. statist. Psychol.*, 7:81–91.

Newman, M. (2004). Detecting community structure in networks. *Eur. Phys. J. B*, 38(2):321–330.

Newman, M. E. J. (2003). The structure and function of complex networks.

Newman, M. E. J. (2006a). Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74.

Newman, M. E. J. (2006b). Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, 103(23):8577–8582.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113–1–15.

Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Adv. in Neural Inform. Process. Systems (NIPS'01)*, 14:83–88.

Nørgaard, L. and Ridder, C. (1994). Rank annihilation factor analysis applied to flow injection analysis with photodiode-array detection. *Chemometrics and Intelligent Laboratory Systems*, 23(1):107–114.

Olshausen, B. A. (2003). Learning sparse, overcomplete representations of time-varying natural images. *Image Processing, ICIP 2003. Proceedings. 2003 International Conference*, 1:41–44.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

Olshausen, B. A. and Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487.

Olshausen, B. A. and Field, David, J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1. *Vision Research*, 37(23):3311–3325.

Osborne, M., Presnell, B., and Turlach, B. (2000). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403.

Paatero, P. (1997). A weighted non-negative least squares algorithm for three-way 'parafac' factor analysis. *Chemom.Intell.Lab.Syst.*, 38:223–242.

Paatero, P. (2000). Construction and analysis of degenerate parafac models. *Journal of Chemometrics*, 14(3):285–299.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

Parra, L., Spence, C., and Vries, B. (1998). Convolutive blind source separation based on multiple decorrelation. *IEEE Workshop on Neural Networks and Signal Processing*, pages 23–32.

Pauca, V. P. P., Piper, J., and Plemmons, R. J. (2006). Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and its Applications*, 416 (2006):29–47.

Perez-Orive, J., Mazor, O., Turner, G., Cassenaer, S., Wilson, R., and Laurent, G. (2002). Oscillations and sparsening of odor representations in the mushroom body. *Science*, 297:359–365.

Petersen, K. B. (2005). *Mean Field ICA*. PhD thesis, Technical University of Denmark, Informatics and Mathematical Modelling.

Pham, D. and Garrat, P. (1997). Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Transaction on Signal Processing Proceedings of NNSP '98*, 45:1712–1725.

Plumbley, M. (2003). Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543.

Pothen., A., Simon, H., and Liou, K.-P. (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11:430–452.

Rasmussen, P. M., Mørup, M., Hansen, L. K., and Arnfred, S. M. (2008). Model order estimation for independent component analysis of epoched eeg signals. *Biosignals 2008 International Conference on Bio-inspired Systems and Signal Processing.*

Reis, M. M., Biloti, D. N., Ferreira, M. M. C., Pessine, F. B. T., and Teixeira, G. M. (2001). Parafac for spectral curve resolution: A case study using total luminescence in human dental tartar. *Appl. Spectrosc. Chemom. Intell. Lab. Syst.*, 55(7):847–851.

Rojas, M. and Steihaug, T. (2002). Large-scale optimization techniques for nonnegative image restorations. *Advanced Signal Processing Algorithms, Architectures, and Implementations XII. Proceedings of the SPIE*, 4791:233–242.

Rosset, S. (2005). Following curved regularized optimization solution paths. pages 1153–1160.

Salakhutdinov, R., Roweis, S., and Ghahramani, Z. (2003). On the convergence of bound optimization algorithms. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 509–516, San Francisco, CA. Morgan Kaufmann Publishers.

Saunders, D. R. (1953). An analytic method for rotation to orthogonal simple structure. *Princeton: Educational Testing Service Research Bulletin*, pages 53–10.

Saunders, M. (2003). Pdco: Primal-dual method for optimization with convex objectives. http://www.stanford.edu/group/SOL/software/pdco/pdco.pdf.

Schmidt, M. N. and Mørup, M. (2006). Nonnegative matrix factor 2-d deconvolution for blind single channel source separation. *Independent Component Analysis and Blind Signal Separation, pages 700-707, 2006*, pages 700–707.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464.

Shaobing, S. C. and Donoho, D. (1994). Basis pursuit. *28th Asilomar conf. Signals, Systems Computers.*

Sidiropoulos, N. D. and Bro, R. (2000). On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics*, 14:229–239.

Smaragdis, P. (2004). Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs. *International Symposium on Independent Component Analysis and Blind Source Separation (ICA)*, 3195:494.

Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180.

Smilde, A., Bro, R., and Geladi, P. (2004). *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley.

Smilde, A. K., Tauller, R., Saurina, J., and Bro, R. (1999). Calibration methods for complex second-order data. *Analytica Chimica Acta*, 398:237–251.

Smith, E. and Lewicki, M. S. (2005). Efficient coding of time-relative structure using spikes. *Neural Computation*, 17:19–45.

Spearman, C. (1904). "general intelligence," objectively determined and measured. *American Journal of Psychology*, 15:201–293.

Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., and Chen, Z. (2005). Cubesvd: a novel approach to personalized web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 382–390, New York, NY, USA. ACM Press.

Syskind, M. P., Larsen, J., Kjems, U., and Parra, L. C. (2007). A survey of convolutive blind source separation methods. *Springer Handbook on Speech Processing and Speech Communication*.

Theunissen, F. (2003). From synchrony to sparseness. *Trends Neurosci.*, 26:61–64.

Thompson, L. and Best, P. (1989). Place cells and silent cells in the hippocampus of freely-behaving rats. *J. Neurosci.*, 9:2382–2390.

Thurstone, L. L. (1947). *Multiple-factor analysis*. Chicago: Univ. Chicago Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Tomasi, G. (2006). *Practical and computational aspects in chemometric data analysis*. PhD thesis, The Royal Veterinary and Agricultural University, Frederiksberg, Denmark.

Tootell, R., Silverman, M., Switkes, E., and De Valois, R. (1982). Deoxyglucose analysis of retinotopic organization in primate striate cortex. *Science*, 218(4575):902–904.

Torkkola, K. (1996). Blind separation of delayed sources based on information-maximization. *Acoustics, Speech, and Signal Processing. ICASSP-96*, 6:3509–3512.

Tryon, R. (1939). *Cluster Analysis*. New York: McGraw-Hill.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.

Vasilescu, M. and Terzopoulos, D. (2005). Multilinear independent components analysis. *Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2005.*

Vasilescu, M. A. O. and Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 447–460, London, UK. Springer-Verlag.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat. Comput.*, 17:395–416.

Wang, D. (2001). Book review: Unsupervised learning - foundations of neural computation edited by geoffrey hinton and terrence j. sejnowski the mit press, 1999. *AI Magazine*, 22:101–102.

Wang, H. and Ahuja, N. (2003). Facial expression decomposition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 958, Washington, DC, USA. IEEE Computer Society.

Wang, K., Begleiter, H., and Porjesz, B. (2000). Trilinear modeling of event-related potentials. *Brain Topography*, 12(4):263–271.

Wang, L., Gordon, M. D., and Zhu, J. (2006). Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. *Sixth IEEE International Conference on Data Mining (ICDM'06)*, pages 690–700.

Welling, M. and Weber, M. (2001). Positive tensor factorization. *Pattern Recogn. Lett.*, 22(12):1255–1261.

Wersing, H., Eggert, J., and Korner, E. (2003). Sparse coding with invariance constraints. *Proc. Int. Conf. Artificial Neural Networks ICANN*, pages 385–392.

Wu, T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat.*, 2(1):224–244.

Yeredor, A. (2003). Time-delay estimation in mixtures. *ICASSP*, 5:237–240.

Yeredor, A. (2005). Blind source separation in the presence of doppler frequency shifts. *ICASSP*, 5:277–280.

Zachary, W. (1977). An information flow model for conflict and fission in small groups. *J. Anthropology Res.*, 33(4):452–473.

Zha, H., Ding, C., Gu, M., He, X., and Simon, H. (2001). Spectral relaxation for k-means clustering. *Adv. Neural Inform. Process. Systems (NIPS'01*, 14:1057–1064.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286.